DMM-32DX-AT User Manual

16-Bit Analog I/O PC/104 Module
with Autocalibration

Rev A.03 March 2011

Copyright © 2011
Diamond Systems Corporation
555 Ellis Street
Mountain View, CA 94043
Tel (650) 810-2500
www.diamondsystems.com

TABLE OF CONTENTS

1. DESCRIPTION 4
2. FEATURES 5
3. DMM-32DX-AT BOARD DRAWING 6
4. I/O HEADERS PINOUT AND PIN DESCRIPTION 7
   - *4.1 Analog I/O Header – J3* *7*
   - *4.2 Digital I/O Header – J4* *8*
   - *4.3 Serial I/O Header – J11* *9*
5. BOARD CONFIGURATION 10
   - *5.1 Base Address* *10*
   - *5.2 Interrupt level* *10*
   - *5.3 DMA level* *10*
   - *5.4 Single-Ended / Differential A/D Channels* *11*
   - *5.5 D/A Configuration* *12*
   - *5.6 Digital I/O Pull-Up / Pull-Down* *12*
   - *5.7 16-Bit Data Bus* *12*
6. I/O REGISTER MAP 13
   - *6.1 I/O Map Description* *13*
   - *6.2 I/O Map Reference – Write* *14*
   - *I/O Map Reference – Read* *15*
   - *6.3 I/O Map Details* *17*
   - *6.4 Page 0: 82C54 Counter/Timer Access* *25*
   - *6.5 Page 1: 82C55 Digital I/O Circuit* *26*
   - *6.6 Page 2: Expanded FIFO Control* *27*
   - *6.7 Page 3: Autocalibration Registers* *27*
   - *6.8 Page 4: dsPIC Interface* *29*
   - *6.9 Page 5: D/A Waveform Generator* *31*
   - *6.10 Page 6: CPLD I/O Window* *32*
   - *6.11 Page 7: D/A Output Channel Control* *32*
7. ENABLING ENHANCED FEATURES 34
   - *7.1 Enabling Enhanced Features (Enhanced Mode)* *34*
   - *7.2 Disabling Enhanced Features (Normal Mode)* *34*
8. ANALOG INPUT RANGES AND RESOLUTION 35
   - *8.1 Unipolar and Bipolar Inputs* *35*
   - *8.2 Input Ranges and Resolution* *35*
   - *8.3 A/D Conversion Formulas* *36*
   - *8.4 Correlation between A/D Code and Input Voltage* *36*
9. PERFORMING AN A/D CONVERSION 38
   - *9.1 Select the input channel or input channel range* *38*
   - *9.2 Select the analog input range* *39*
   - *9.3 Wait for the analog circuit to settle* *39*
   - *9.4 Start an A/D conversion on the current channel* *39*
   - *9.5 Wait for the conversion to finish* *39*
   - *9.6 Read the A/D data* *39*
   - *9.7 Convert the numerical data to a meaningful value* *39*
10. A/D SAMPLING METHODS 40
    - *10.1 Sampling Modes* *40*
    - *10.2 FIFO Description* *41*
    - *10.3 Scan Sampling* *41*
    - *10.4 Sequential Sampling* *41*
11. HOW TO PERFORM A/D CONVERSIONS USING INTERRUPTS 42
12. ANALOG OUTPUT RANGES AND RESOLUTION 43
    - *12.1 Description* *43*
    - *12.2 Resolution* *43*
    - *12.3 Full-Scale Range Selection* *43*

13. GENERATING AN ANALOG OUTPUT 44
13.1 Compute the D/A code for the desired output voltage *44*
13.2 Compute the LSB and MSB values *44*
13.3 Add the channel number to the MSB *44*
13.4 Set D/A Simultaneous Update bit *44*
13.5 Write the LSB and MSB to the board *45*
13.6 Monitor the DACBUSY status bit *45*
14. D/A WAVEFORM GENERATOR 46
14.1 Description *46*
14.2 Programming the D/A wave form generator *46*
14.3 Enable enhanced features *46*
14.4 Reset D/A wave form pointer *46*
14.5 Latch D/A value *46*
14.6 Store D/A values into buffer *46*
14.7 Setup D/A wave form settings *47*
14.8 Start D/A waveform generator *47*
15. AUTOCALIBRATION 48
16. DIGITAL I/O OPERATION 50
16.1 Main Digital I/O on J4: Internal 82C55 Circuit *50*
16.2 Digital I/O Configuration Register *51*
16.3 Mode 0 Digital I/O *52*
16.4 Mode 1 Digital I/O With Handshaking *52*
16.5 Auxiliary Digital I/O on J3 ..... *53*
17. COUNTER/TIMER OPERATION 54
17.1 Counter/Timer Features and Configuration Options *54*
17.2 Counter/Timer Configuration *54*
17.3 Counter/Timer Access and Programming *54*
18. SPECIFICATIONS 55

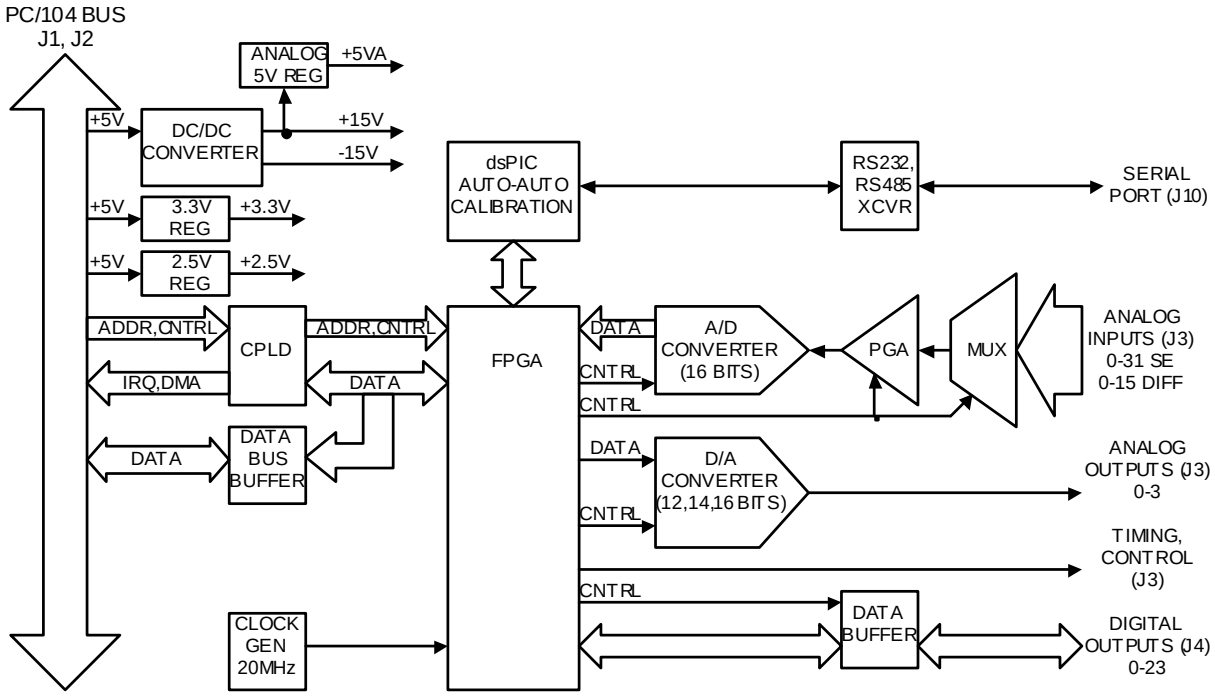
1. DESCRIPTION

DMM-32DX-AT is a PC/104-format data acquisition board with a full set of analog and digital I/O features. It offers 32 analog inputs with 16-bit resolution and programmable input range; 250,000 samples per second maximum sampling rate with FIFO operation; 4 analog outputs with 16-bit (or 12-bit) resolution; user-adjustable analog output ranges; 31 lines of digital I/O; one 32-bit counter/timer for A/D conversion and interrupt timing; and one 16-bit counter/timer for general purpose use.

The DMM-32DX-AT is designed to be a fully backwards-compatible upgrade for the DMM-32-AT board. In addition to all DMM-32-AT features, the DMM-32DX-AT includes the following upgrades:

- 1024-sample FIFO for A/D samples versus. 512 samples on DMM-32-AT
- 1024-sample data buffer for D/A waveform generation
- Software reprogrammable FPGA and dsPIC microcontroller for future feature enhancements
- Ability to issue commands to DMM-32DX-AT through a serial port
- Patented auto-autocalibration feature that provides fully autonomous calibration in hardware

BLOCK DIAGRAM



2. FEATURES

Analog Inputs

- 32 input channels, 16-bit resolution
- May be configured as 32 single-ended, 16 differential, or 16 SE + 8 DI
- Programmable gain, range, and polarity on inputs
- 250,000 samples per second maximum sampling rate
- 1024-sample FIFO for reduced interrupt overhead
- Autocalibration of all input ranges under software control
- Patented hardware-implemented auto-autocalibration

Analog Outputs

- 4 analog output channels with 16-bit standard, 12-bit optional resolution, 5mA max output current
- Multiple fixed full-scale output ranges, including unipolar and bipolar ranges
- Programmable full-scale range
- Patented hardware-implemented auto-autocalibration
- 1024-sample FIFO for D/A wave form generation
- 250 KHz maximum update rate
- Simultaneous update for all 4 channels capability

Digital I/O

- 24 bi-directional lines using integrated 8255-type circuit
- Buffered I/O for enhanced current drive
- Handshaking controls enable external latching of data as well as interrupt operation
- User-configurable pull-up / pull-down resistors
- 7 additional I/O lines are fixed direction with programmable functions

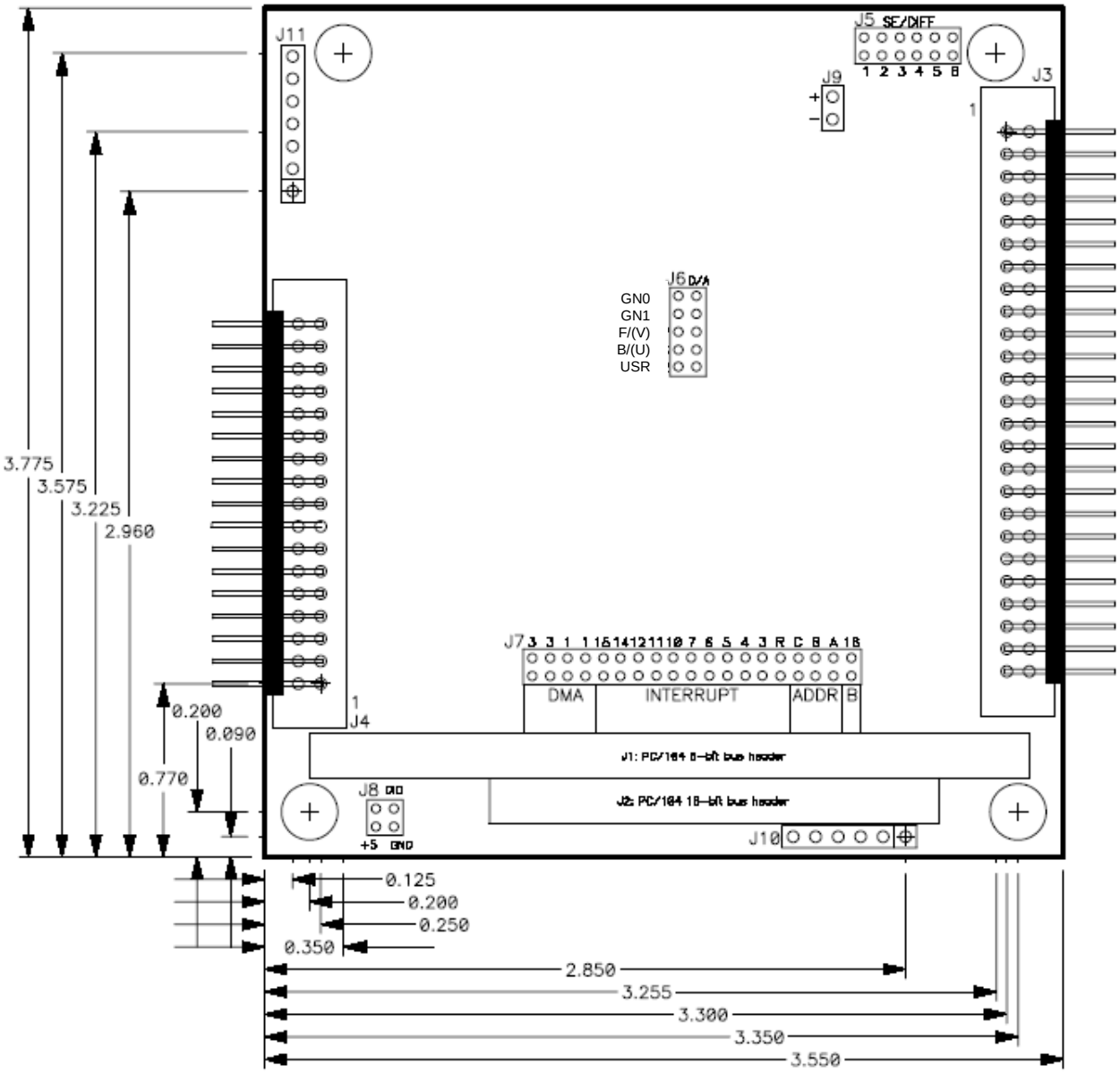
Counter/Timers and A/D Triggering

- 1 32-bit counter/timer for A/D pacer clock and interrupt operation timing
- 1 16-bit general purpose counter/timer
- Programmable input sources for each counter/timer
- External A/D triggering and gating inputs
- Multiple-board synchronization capability using A/D convert pulse out and external trigger in
- Interrupts may be generated by counter/timer

Miscellaneous

- Extended temperature –40^{o}C to +85^{o}C operation
- No trimpots or user adjustments required for calibration. Auto autocalibration will automatically adjust the A/D without user input. Calibration time for all modes is approximately 2 seconds.
- Auto autocalibration of one A/D mode using the onboard dsPIC requires approx 0.5 seconds.

3. DMM-32DX-AT BOARD DRAWING



I/O Connectors and Features

J1	PC/104 8-bit bus header
J2	PC/104 16-bit bus header (only used for interrupt level)
J3	Analog I/O header (includes trigger and ctr/timer signals)
J4	Digital I/O header
5	Analog input single-ended / differential configuration
J6	D/A unipolar / bipolar / full-scale range configuration
J7	Base address / DMA level / interrupt level / bus width
J8	Digital I/O pull-up / pull-down configuration
J9	Test connector; not used in normal operation
J10	JTAG programming cable; not used in normal operation
J11	Auxiliary power / serial connector
LED	User-programmable LED

4. I/O HEADERS PINOUT AND PIN DESCRIPTION

4.1 Analog I/O Header – J3

Diamond-MM-32DX-AT provides a 50-pin header on the right edge of the board labeled J3 for all I/O relating to analog functions. Pin 1 is in the upper left corner.

J3: Analog I/O Header

AGND	1	2	AGND
Vin 0 / 0+	3	4	Vin 16 / 0-
Vin 1 / 1+	5	6	Vin 17 / 1-
Vin 2 / 2+	7	8	Vin 18 / 2-
Vin 3 / 3+	9	10	Vin 19 / 3-
Vin 4 / 4+	11	12	Vin 20 / 4-
Vin 5 / 5+	13	14	Vin 21 / 5-
Vin 6 / 6+	15	16	Vin 22 / 6-
Vin 7 / 7+	17	18	Vin 23 / 7-
Vin 8 / 8+	19	20	Vin 24 / 8-
Vin 9 / 9+	21	22	Vin 25 / 9-
Vin 10 / 10+	23	24	Vin 26 / 10-
Vin 11 / 11+	25	26	Vin 27 / 11-
Vin 12 / 12+	27	28	Vin 28 / 12-
Vin 13 / 13+	29	30	Vin 29 / 13-
Vin 14 / 14+	31	32	Vin 30 / 14-
Vin 15 / 15+	33	34	Vin 31 / 15-
Vout 3	35	36	Vout 2
Vout 1	37	38	Vout 0
Vref Out	39	40	Agnd
A/D Convert	41	42	Ctr 2 Out / Dout 2
Dout 1	43	44	Ctr 0 Out / Dout 0
Extclk / Din 3	45	46	Extgate / Din 2
Gate 0 / Din 1	47	48	Clk 0 / Din 0
+5V	49	50	Dgnd

Signal Name	Definition
Vin 15/15+ ~ Vin 0/0+	Analog input channels 15 - 0 in single-ended mode; High side of input channels 15 - 0 in differential mode
Vin 31/16- ~ Vin 15/0-	Analog input channels 31 - 16 in both single-ended mode; Low side of input channels 15 - 0 in differential mode
Vref Out	Precision +5V signal for reference only. Do not use for power.
Vout 0 - 3	16-bit standard, (12-bit optional) analog output channels
A/D Convert	A/D convert signal output; can be used to synchronize multiple boards
Dout 2 – Dout 0	Digital output port with counter/timer functions
Din 3 – Din 0	Digital input port with counter/timer and external trigger functions
Extclk	External A/D trigger input; Also used for digital interrupt (DINT) input
Extgate	Pin to control gating of Ctrs 1& 2 for A/D timing
Gate 0	Pin to control gating of Ctr 0
Clk 0	Input source to Ctr 0
+5V	Connected to PC/104 bus power supply
Agnd	Analog ground; connected to digital ground at a single point at DC/DC converter PS1 on board
Dgnd	Digital ground; connected to PC/104 bus ground

4.2 Digital I/O Header – J4

Diamond-MM-32DX-AT provides a 34-pin header on the left edge of the board labeled J4 for the 24 8255-type digital I/O lines. Pin 1 is in the lower left corner.

J4: Digital I/O Header

A7	1	2	A6
A5	3	4	A4
A3	5	6	A2
A1	7	8	A0
B7	9	10	B6
B5	11	12	B4
B3	13	14	B2
B1	15	16	B0
C7	17	18	C6
C5	19	20	C4
C3	21	22	C2
C1	23	24	C0
Latch	25	26	Ack
NC	27	28	NC
(RS-232) Tx	29	30	Rx (RS-232)
(RS-485) Rx/Tx +	31	32	Rx/Tx - (RS-485)
+5V	33	34	Dgnd

Signal Name	Definition
A7 – A0	Digital I/O port A
B7 – B0	Digital I/O port B
C7 – C0	Digital I/O port C
Latch	Latch control input; active high
Ack	Acknowledge output for interrupt-based I/O; active high
RS-232 Rx/Tx	RS-232 transceiver terminals to communicate with onboard dsPIC
RS-485 Rx/Tx +/-	RS-485 transceiver terminals to communicate with onboard dsPIC
+5V	Connected to PC/104 bus +5V power supply
Dgnd	Digital ground; connected to PC/104 bus ground

Note: The operation of digital I/O Latch and Ack signals is detailed in Chapter 16.

4.3 Serial I/O Header – J11

Diamond-MM-32DX-AT provides a 7-pin right-angle friction-lock header on the left edge of the board labeled J11 for the auxiliary serial and power connections.

The serial connections provide an alternate means to communicate with the board instead of the PC/104 bus. Only one serial protocol may be used at a time. The board will respond to either protocol without any configuration required. The serial port may also be used when the board is installed in a PC/104 stack.

The power pins may be used to provide power to the board in standalone operation (not used in a PC/104 stack). When the board is installed in a PC/104 stack, the +5V pin may be used as a source of +5V for auxiliary devices (maximum current 1 Amp) but should NOT be used for power input.

J11: Serial I/O Header

1	+Vin
2	+5V
3	Digital GND
4	RS-485: (–)
5	RS-485: (+)
6	RS-232: Tx
7	RS-232: Rx

Signal Name	Definition
+Vin	Unregulated Power Input. This allows the board to operate in standalone mode, fed through an unregulated power supply ranging from 7 to 12VDC. This pin requires an optional voltage regulator installed on the bottom side of the board. Do not attempt to use this pin unless the regulator is installed.
+5V	This pin can be connected to a regulated +5VDC power supply to power the board for standalone operation.
Digital GND	Connected to the digital ground plane of the board.
RS-232 Rx/Tx	RS-232 receive/transmit lines to communicate with onboard dsPIC
RS-485 (+)/(–)	RS-485 receive/transmit lines to communicate with onboard dsPIC

5. BOARD CONFIGURATION

Refer to the Drawing of Diamond-MM-32DX-AT on page 6 for locations of the configuration items mentioned here.

5.1 Base Address

Each board in a PC/104 system must have its own unique block of addresses that does not overlap with any other board in the system or feature on the CPU. The lowest address of this address block is called the base address. Diamond-MM-32DX-AT's base address is set with 3 pairs of pins marked “ADDR” on pin header **J7**, located near the PC/104 bus connectors. The table below lists the 8 possible jumper configurations and the corresponding base addresses.

Base Address Configuration

Base Address		Pin Header J7 Configuration			
Hex	Decimal	C	B	A	
140	320	Installed	Installed	Installed	
340	832	Installed	Installed	Open	
100	256	Installed	Open	Installed	
180	384	Installed	Open	Open	
200	512	Open	Installed	Installed	
280	640	Open	Installed	Open	
300	768	Open	Open	Installed	**Default Setting**
380	896	Open	Open	Open	

5.2 Interrupt level

Interrupts are used for hardware I/O operations that are independent of normal program flow. Diamond-MM-32DX-AT can be set up to generate interrupts under several circumstances. The most common use of interrupts is to transfer A/D data from the board to system memory during high-speed A/D sampling. The board can also generate interrupts to transfer digital data into or out of the board, as well as at regular intervals according to a programmable timer on the board. Individual control bits are used to enable each type of interrupt.

Jumper block J7 contains pins for selecting the interrupt level. To set the desired level, install a jumper under that level’s IRQ number and also in the R position. The R position connects a 1KΩ pull-down resistor to the selected IRQ line to allow the board to share the IRQ level with another board in the system. Note that only one pulldown resistor should be installed on any IRQ level. If you have another board in the system using the same IRQ level as Diamond-MM-32DX-AT, and that board has the pulldown resistor already configured, then remove the jumper from the R position on J7.

5.3 DMA level

Jumper block J7 contains pins for selecting the DMA level between channels 1 and 3. To select channel, 1 both jumpers labeled “1” must be on. To select channel 3, both jumpers labeled “3” must be on. DMA is supported in the hardware, however it is not currently supported in Diamond Systems’ Universal Driver software.

On boards without FIFOs or memory buffers, DMA is required to support high-speed sampling at rates above the maximum sustainable interrupt rate, which may vary from 1,000 to 20,000 depending on the CPU and operating system. However, DMM-32DX-AT contains a 1024-sample FIFO for A/D data that allows the interrupt rate to be much slower than the sample rate. The board can support full-speed sampling at up to 250,000 samples per second without the use of DMA.

5.4 Single-Ended / Differential A/D Channels

The input channels on DMM-32DX-AT can be configured as 32 single-ended, 16 differential, or 16 single-ended + 8 differential. Four different configurations are possible as described below.

A **single-ended input** is a single-wire input (plus ground) that is measured with reference to the board's analog ground. In order for the measurement to be accurate, the board's ground must be at the same potential as the source signal's ground. Usually this is accomplished by connecting the two grounds together at some point, for example by connecting to one of the analog ground pins on the I/O header J3.

A **differential input** is a two-wire input (plus ground) that is measured by subtracting the low input from the high input. This type of connection offers two advantages: It allows for greater noise immunity, because the noise, which is present in equal amounts and equal phase on both the high and low inputs, is subtracted out when the low input is subtracted from the high input; and it allows for the signal to float away from ground. Normally the ground of the signal source is still connected to the ground on the A/D board in order to keep the signal from straying out of the common mode range of the A/D board's input circuitry.

To configure the input channels, set jumpers in jumper block J5 according to the table below. For safety reasons, do not modify J5 while the board is powered on. The corresponding channel numbering on the I/O header J3 is shown in drawings A-D (only the first 17 rows are shown; the remaining rows are the same as shown on page 7).

A/D Channel Mode Configuration

Configuration	I/O Header Drawing	Jumper Settings 1	2	3	4	5	6
0-31 SE	A	In	In	Out	Out	Out	Out
0-15 DI	B	Out	Out	In	In	In	In
0-7 DI, 8-15 SE, 24-31 SE	C	Out	In	In	Out	In	Out
0-7 SE, 8-15 DI, 16-23 SE	D	In	Out	Out	In	Out	In

A

Agnd	1	2	Agnd
0	3	4	16
1	5	6	17
2	7	8	18
3	9	10	19
4	11	12	20
5	13	14	21
6	15	16	22
7	17	18	23
8	19	20	24
9	21	22	25
10	23	24	26
11	25	26	27
12	27	28	28
13	29	30	29
14	31	32	30
15	33	34	31
	35-50 Same as p. 7		

B

Agnd	1	2	Agnd
0+	3	4	0-
1+	5	6	1-
2+	7	8	2-
3+	9	10	3-
4+	11	12	4-
5+	13	14	5-
6+	15	16	6-
7+	17	18	7-
8+	19	20	8-
9+	21	22	9-
10+	23	24	10-
11+	25	26	11-
12+	27	28	12-
13+	29	30	13-
14+	31	32	14-
15+	33	34	15-
	35-50 Same as p. 7		

C

Agnd	1	2	Agnd
0+	3	4	0-
1+	5	6	1-
2+	7	8	2-
3+	9	10	3-
4+	11	12	4-
5+	13	14	5-
6+	15	16	6-
7+	17	18	7-
8	19	20	24
9	21	22	25
10	23	24	26
11	25	26	27
12	27	28	28
13	29	30	29
14	31	32	30
15	33	34	31
	35-50 Same as p. 7		

D

Agnd	1	2	Agnd
0	3	4	16
1	5	6	17
2	7	8	18
3	9	10	19
4	11	12	20
5	13	14	21
6	15	16	22
7	17	18	23
8+	19	20	8-
9+	21	22	9-
10+	23	24	10-
11+	25	26	11-
12+	27	28	12-
13+	29	30	13-
14+	31	32	14-
15+	33	34	15-
	35-50 Same as p. 7		

5.5 D/A Configuration

The four analog outputs on DMM-32DX-AT can be set individually or all at once to operate in bipolar (both + and –) or unipolar (+ only) output voltage ranges with the full-scale output range set to 5V, 10V, or programmable. The maximum output current on any channel is 5mA. Current outputs such as 0-20mA outputs are not supported. Moreover, the hardware jumper settings can be overridden by software by setting the bit D/A JPOVRD when the board is operating in maximum mode.

On power-up, the DACs are configured to reset to mid-scale (0V in bipolar mode) or zero scale (0V in unipolar mode) so that the DACs power up to 0V.

In programmable mode, the full-scale output voltage can be set anywhere from 0V to 10V in software. You must use the Universal Driver software to set programmable D/A range, as it requires calibration to fine-tune the setting to the desired value. The Diamond-MM-32DX-AT demo package includes a section showing how to set the D/A range in software.

To configure the analog output range, set jumper block J6 according to the tables below. The first two positions are used for the output range, the third position is to determine the output polarity and the fourth position is to determine a fixed or variable full-scale output. The use of the 5th position can be defined by the user for uses that do or do not involve the D/A converter.

Placing the jumper in position grounds the corresponding control signal. For example, inserting the B/(U) jumper sets the board to unipolar mode.

J6: Analog Output Configuration

	Jumper Settings			
Output Range	**GN1**	**GN0**	**B/(U)**	**F/(V)**
±2.5V	In	In	Out	Out
±5V	In	Out	Out	Out
±10V	Out	In	Out	Out
0-5V	In	In	In	Out
0-10V	In	Out	In	Out
Programmable*	X	X	X	In

* *Programmable mode requires use of driver software to set and calibrate range.*

5.6 Digital I/O Pull-Up / Pull-Down

The 24 digital I/O lines on I/O header J4 are connected to 10K resistors that can be configured for either pull-up or pull-down. All resistors are configured in the same way. Jumper block J8 in the lower left corner of the board is used for configuration. To set the pull direction, install a jumper above the mark **+5** or **GND** as desired. The +5 and Ground signals are wired to opposite corners of J8 to prevent accidentally shorting out the power supply by inserting the jumper incorrectly. **DO NOT install jumpers in both +5 and GND locations simultaneously.**

5.7 16-Bit Data Bus

The board can be configured for 16-bit read operations when reading the A/D data. To do this, install a jumper in the “16” location on J7. A 16-bit transfer will only occur during a 16-bit read instruction from the base address (A/D data) when a jumper is in the “16” position and the board is in a 16-bit bus (both PC/104 J1 and J2 connectors are connected to the CPU). Otherwise the A/D board and host CPU will ignore the 16-bit setting and/or instruction and convert the 16-bit operation into two 8-bit read operations from Base + 0 and Base + 1.

6. I/O REGISTER MAP

Diamond-MM-32DX-AT occupies 16 bytes in the system I/O address space. Direct register access is not required if you are using the Universal Driver™ software that ships with the board. The driver handles all board access and provides a high-level set of functions to simplify programming. The information presented here and in the next chapter is intended to provide a detailed description of the board's features and operation, as well as for programmers who are not using the Universal Driver software.

The DMM-32DX-AT FPGA I/O register map, while remaining backwards compatible with the DMM-32-AT, offers more features to the user. New features that did not exist in the DMM-32-AT are marked **boldface**,or noted as an **enhanced feature**.

Registers 12 – 15 provide a window into several pages for access to additional registers without requiring additional I/O address space. Page selection is done via control bits in register 8.

6.1 I/O Map Description

Base +	Write Function	Read Function
	Main Registers	
0	Start A/D Conversion	A/D LSB (bits 7-0)
1	Auxiliary digital output	A/D MSB (bits 15-8)
2	A/D low channel register	A/D low channel read-back
3	A/D high channel register	A/D high channel read-back
4	D/A LSB register	Auxiliary digital input
5	D/A MSB + channel register	
6	FIFO threshold register	FIFO depth register
7	FIFO control register	FIFO status register
8	Miscellaneous and page control register	Status register
9	Operation control register	Operation status register
10	Counter/timer control register	Counter/timer control read-back
11	Analog configuration register	Analog configuration read-back
	Page 0: 82C54 Counter/Timer Access	
12	Counter 0 data	Counter 0 data read-back
13	Counter 1 data	Counter 1 data read-back
14	Counter 2 data	Counter 2 data read-back
15	82C54 control register	82C54 control register read-back
	Page 1: 82C55-Type Digital I/O	
12	Port A Output	Port A Input
13	Port B Output	Port B Input
14	Port C Output	Port C Input
15	DIO control register	DIO control register read-back
	Page 2: FIFO Control (Enhanced Feature Page)	
12	Expanded FIFO depth register	Expanded FIFO depth read-back
13		
14		
15		
	Page 3: Autocalibration Registers	
12	EEPROM/TrimDAC data latch	EEPROM/TrimDAC data read-back
13	EEPROM/TrimDAC address latch	EEPROM/TrimDAC address read-back
14	EEPROM/TrimDAC control register	EEPROM/TrimDAC status register
15	Special features unlock register0	FPGA revision code

Page 4: dsPIC Interface (Enhanced Feature Page)		
12	dsPIC data latch	dsPIC data latch
13	dsPIC address latch	dsPIC address latch
14	dsPIC control register	dsPIC status register
15	dsPIC programming control	dsPIC programming status
Page 5: D/A Waveform Generator (Enhanced Feature Page)		
12	Waveform address latch (LSB)	Waveform address latch (LSB)
13	Waveform address latch (MSB)	Waveform address latch (MSB)
14	Waveform configuration register	Waveform configuration read-back
15	Waveform command register	
Page 6: CPLD I/O Window (Enhanced Feature Page)		
12		
13	This page is a window to the CPLD I/O.	
14	This page should not be accessed under normal operation.	
15		
Page 7: D/A Channel Control (Enhanced Feature Page, When Enabled by DAC_CONT)		
12	D/A Channel A Control	D/A Channel A Control
13	D/A Channel B Control	D/A Channel B Control
14	D/A Channel C Control	D/A Channel C Control
15	D/A Channel D Control	D/A Channel D Control

6.2 I/O Map Reference – Write

Base +	7	6	5	4	3	2	1	0
Main Registers								
0	--- Start A/D Conversion ---							
1					**LED**	DOUT2	DOUT1	DOUT0
2				L4	L3	L2	L1	L0
3				H4	H3	H2	H1	H0
4	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
5	DACH1	DACH0	DASIM	DAGEN	DA11	DA10	DA9	DA8
6	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1
7					FIFOEN	SCANEN	FIFORST	
8			RESETA	RESETD	INTRST	**P2**	P1	P0
9	ADINTE	DINTE	TINTE	RSVD1	**DMAEN**		CLKEN	CLKSEL
10	FREQ12	FREQ0	OUT2EN	OUT0EN	RSVD	GT0EN	SRC0	GT12EN
11			**SCINT1**	**SCINT0**	RANGE	ADBU	G1	G0
Page 0: 82C54 Counter/Timer Access								
12	Ctr0D7	Ctr0D6	Ctr0D5	Ctr0D4	Ctr0D3	Ctr0D2	Ctr0D1	Ctr0D0
13	Ctr1D7	Ctr1D6	Ctr1D5	Ctr1D4	Ctr1D3	Ctr1D2	Ctr1D1	Ctr1D0
14	Ctr2D7	Ctr2D6	Ctr2D5	Ctr2D4	Ctr2D3	Ctr2D2	Ctr2D1	Ctr2D0
15	SC1	SC0	RW1	RW0	M2	M1	M0	BCD
Page 1: 82C55-Type Digital I/O								
12	A7	A6	A5	A4	A3	A2	A1	A0
13	B7	B6	B5	B4	B3	B2	B1	B0
14	C7	C6	C5	C4	C3	C2	C1	C0
15	1	ModeC	ModeA	DIRA	DIRCH	ModeB	DIRB	DIRCL
Page 2: FIFO Control (Enhanced Feature Page)								
12								FT9
13								
14								
15								

Page 3: Autocalibration Registers								
12	D7	D6	D5	D4	D3	D2	D1	D0
13		A6	A5	A4	A3	A2	A1	A0
14	EE_EN	EE_RW	RUNCAL	CMUXEN	TDACEN			
15	--FPGA Feature Unlock Register--							
Page 4: dsPIC Interface (Enhanced Feature Page)								
12	PICD7	PICD6	PICD5	PICD4	PICD3	PICD2	PICD1	PICD0
13	I2CBSY			PICA4	PICA3	PICA2	PICA1	PICA0
14				ACHOLD	ACREL	PICRST	ACABT	ACTRIG
15	PSTART	PSTOP	PGDOUT	PGDIN	PGDW1	PGDW0	PGCW1	PGCW0
Page 5: D/A Waveform Generator (Enhanced Feature Page)								
12	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0
13							DACA9	DACA8
14	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0
15					WGINC	WGRST	WGPS	WGSTRT
Page 6: CPLD I/O Window (Enhanced Feature Page)								
12	This page is a window to the CPLD I/O and should not be accessed under normal operation.							
13								
14								
15								
Page 7: D/A Channel Control (Enhanced Feature Page, When Enabled by DAC_CONT)								
12	D/A BIT 07	D/A BIT 06	D/A BIT 05	D/A BIT 04	D/A BIT 03	D/A BIT 02	D/A BIT 01	D/A BIT 00
13	D/A BIT 15	D/A BIT 14	D/A BIT 13	D/A BIT 12	D/A BIT 11	D/A BIT 10	D/A BIT 09	D/A BIT 08
14		D/A ONE/ALL	REG SELECT	D/A JPOVRD	D/A FX/VR	D/A GAIN1	D/A GAIN0	D/A POL
15								

I/O Map Reference – Read

Base +	7	6	5	4	3	2	1	0
Main Registers								
0	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
1	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8
2				L4	L3	L2	L1	L0
3				H4	H3	H2	H1	H0
4	DACBUSY	CALBUSY	**ACACT**	**USRDEF**	DIN3	DIN2	DIN1	DIN0
5								FD9
6	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1
7	EF	TF	FF	OVF	FIFOEN	SCANEN	PAGE1	PAGE0
8	STS	S/D1	S/D0	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
9	ADINT	DINT	TINT	X	**DMAEN**	**P2**	CLKEN	CLKSEL
10	FREQ12	FREQ0	OUT2EN	OUT0EN	RSVD	GT0EN	SRC0	GT12EN
11	WAIT	RSVD	**SCINT1**	**SCINT0**	RANGE	ADBU	G1	G0
Page 0: 82C54 Counter/Timer Access								
12	Ctr0D7	Ctr0D6	Ctr0D5	Ctr0D4	Ctr0D3	Ctr0D2	Ctr0D1	Ctr0D0
13	Ctr1D7	Ctr1D6	Ctr1D5	Ctr1D4	Ctr1D3	Ctr1D2	Ctr1D1	Ctr1D0
14	Ctr2D7	Ctr2D6	Ctr2D5	Ctr2D4	Ctr2D3	Ctr2D2	Ctr2D1	Ctr2D0
15	SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Page 1: 82C55-Type Digital I/O								
12	A7	A6	A5	A4	A3	A2	A1	A0
13	B7	B6	B5	B4	B3	B2	B1	B0
14	C7	C6	C5	C4	C3	C2	C1	C0
15	1	ModeC	ModeA	DIRA	DIRCH	ModeB	DIRB	DIRCL
Page 2: FIFO Control (Enhanced Feature Page)								
12								FD9
13								
14								
15								
Page 3: Autocalibration Registers								
12	D7	D6	D5	D4	D3	D2	D1	D0
13		A6	A5	A4	A3	A2	A1	A0
14		TDBUSY	EEBUSY	CMUXEN	TDACEN			
15	--- FPGA Revision Code ---							
Page 4: dsPIC Interface (Enhanced Feature Page)								
12	PICD7	PICD6	PICD5	PICD4	PICD3	PICD2	PICD1	PICD0
13	I2CBSY			PICA4	PICA3	PICA2	PICA1	PICA0
14				ACHOLD	PICPRST	ACERR	ACACT	PICBSY
15								PGDR
Page 5: D/A Waveform Generator (Enhanced Feature Page)								
12	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0
13							DACA9	DACA8
14	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0
15								
Page 6: CPLD I/O Window (Enhanced Feature Page)								
12	This page is a window to the CPLD I/O and should not be accessed under normal operation.							
13								
14								
15								
Page 7: D/A Channel Control (Enhanced Feature Page, When Enabled by DAC_CONT)								
12	D/A BIT 07	D/A BIT 06	D/A BIT 05	D/A BIT 04	D/A BIT 03	D/A BIT 02	D/A BIT 01	D/A BIT 00
13	D/A BIT 15	D/A BIT 14	D/A BIT 13	D/A BIT 12	D/A BIT 11	D/A BIT 10	D/A BIT 09	D/A BIT 08
14		J_DAC SZ1	J_DAC SZ0	D/A JPOVRD	D/A FX/VR	D/A GAIN1	D/A GAIN0	D/A POL
15								

6.3 I/O Map Details

This section describes the location and general behavior of specific bits in each I/O map register. In all register definitions below, a bit named X is not defined and serves no function.

Base + 0 Write Start A/D Conversion

Writing to Base + 0 starts an A/D conversion, unless a conversion is already in progress (AD_BUSY high.) The value written does not matter. Writing to Base + 0 will start an A/D conversion even if the board is set up for interrupt, DMA, or external trigger mode.

Base + 0 Read A/D LSB

Bit No.	7	6	5	4	3	2	1	0
Name	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

AD7-0 A/D data bits 7 - 0; AD0 is the LSB

Base + 1 Write Auxiliary Digital Output

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	**LED**	DOUT2	DOUT1	DOUT0

DOUT2-0 Auxiliary digital output bits on analog I/O header J3. Two pins also serve as optional counter outputs based on control register bits at Base + 10:

DOUT2 J3 pin 42. Counter 2 output when OUT2EN = 1 (Base + 10 bit 5).

DOUT1 J3 pin 43

DOUT0 J3 pin 44. Counter 0 output when OUT0EN = 1 (Base + 10 bit 4).

LED This bit toggles the onboard user LED. 1 = on; 0 = off.

Base + 1 Read A/D MSB

Bit No.	7	6	5	4	3	2	1	0
Name	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8

AD15 - 8 A/D data bits 15 - 8; AD15 is the MSB

Base + 2 Read/Write A/D Low Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	L4	L3	L2	L1	L0

L4-0 The low channel number setting in the A/D channel scan range. Channel numbers range from 0 to 31 in single-ended mode. Writing to this register updates the current channel internal register.

Base + 3 Read/Write A/D High Channel Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	H4	H3	H2	H1	H0

Definitions:

H4-0 The high channel number setting in the A/D channel scan range. Channel numbers range from 0 to 31 in single-ended mode.

Base + 4 Write DAC LSB

Bit No.	7	6	5	4	3	2	1	0
Name	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

This register is used to write the lower bits of a 12-bit data value to the D/A. If data is written to page 7 base + 13, then the next time base + 5 is written, the FPGA will send the data from page 7 base + 12 & 13 to the channel selected by the channel bits in base + 5. After one such cycle, the FPGA will revert to using base + 4 & 5 for the D/A data until page 7 base + 13 is written to again. If page 7 base + 13 has not been written to since the last time base + 5 was written, then when base + 5 is written, the FPGA will assemble the D/A data packet using the data in base + 4 & 5.

Definitions:

DA7 - 0 D/A data bits 7 - 0 for the channel currently being accessed. This register is a holding register. Writing to it does not affect any D/A channel until the MSB is written. When the MSB is written (see below, Base + 5), the value written to that register, along with the value written to this register, are simultaneously written to the D/A chip's load register for the selected channel. See Base+5/write for more details.

Base + 4 Read Status / Auxiliary digital inputs

Bit No.	7	6	5	4	3	2	1	0
Name	DACBUSY	CALBUSY	**ACACT**	**USRDEF**	DIN3	DIN2	DIN1	DIN0

Definitions:

DIN3-0 Auxiliary digital inputs on analog I/O header J3. These pins have multiple functions based on control bits at Base + 9 and Base + 10:

DIN3 J3 pin 45. External A/D clock when CLKSEL = 1 (Base + 9 bit 0)

DIN2 J3 pin 46. Gate for counters 1 and 2 when GT12EN = 1 (Base + 10 bit 0)

DIN1 J3 pin 47. Gate for counter 0 when GT0EN = 1 (Base + 10 bit 2)

DIN0 J3 pin 48. Clock for counter 0 when SRC0 = 1 (Base + 10 bit1)

DACBUSY The D/A serial transfer is in progress. Do not attempt to write to the D/A converters at Base + 4 or Base + 5 while this bit is high. This bit must be checked before any write to these registers.

CALBUSY Calibration is in progress or EEPROM is being accessed. Do not attempt calibration or EEPROM access while this bit is high. This bit must be checked before any calibration or EEPROM operation is attempted.

ACACT This is a copy of the value found at Page 4, Base+14, bit 1. It is mirrored at this location to provide a page-independent means of seeing the AC status, since AC uses Page 3.

USRDEF User Defined Jumper Input. Reports the state of the option jumper signal J_USR_DEF.

Base + 5 Write DAC MSB + Channel No.

Bit No.	7	6	5	4	3	2	1	0
Name	DACH1	DACH0	DASIM	DAGEN	DA11	DA10	DA9	DA8

This register is used to write the upper bits of a 12-bit data value to the D/A and to select the D/A channel. If data is written to page 7 base + 13, then the next time base + 5 is written, the FPGA will send the data from page 7 base + 12 & 13 to the channel selected by the channel bits in base + 5. After one such cycle, the FPGA will revert to using base + 4 & 5 for the D/A data until page 7 base + 13 is written to again. If page 7 base + 13 has not been written to since the last time base + 5 was written, then when base + 5 is written, the FPGA will assemble the D/A data packet using the data in base + 4 & 5.

Definitions:

DA11 – 8 — D/A bits 15 - 8 for the selected output channel; DA15 is the MSB for a 16-bit D/A output, DA13 is the MSB for a 14-bit D/A output and DA11 is the MSB for a 12-bit D/A output (when optional 12-bit D/A converter is installed). Bits 15-12 are enabled when P2 is set to '1'

DACH1-0 — Binary number of the D/A channel, 3 – 0, when P2 is set to '0'

DASIM — D/A simultaneous update when P2 is set to '0'. If DASIM=1 when writing to this register, the D/A conversion is "latched", i.e. the 16-bit (or 12-bit) value will be loaded into the D/A converter, but the output will not change until this register is written to again with DASIM set to 0, at which point all latched D/A channels written to previously will update. Note that this is an enhanced feature. If enhanced features are disabled, DASIM will always be considered a '0' for backwards compatibility, causing D/A outputs to update on every write to this register.

DAGEN — If this bit is '1' no data is transferred to the DAC chip. This is used in conjunction with the D/A waveform generator to store the DAC code that will be written into the waveform memory block. If enhanced features are disabled, this bit is always considered '0' and the data will be transferred to the DAC chip.

Base + 5 Read Update All D/A Channels

Reading from this address causes all 4 D/A channels to update with the values loaded into their load registers. Any channel which has had a new value written to it since the last update command will switch to its new value. Any channel which has not had a new value written will maintain its present value without glitching.

Base + 6 Read / Write FIFO Depth Register

Bit No.	7	6	5	4	3	2	1	0
Name	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1

FT8-1 — FIFO threshold. This is the level at which the board will generate an interrupt request when the FIFO is enabled (FIFOEN = 1 in Base + 7). Note that the value written is shifted by 1 bit, i.e. divided by 2. For example, if you want a FIFO threshold of 256 samples, write a 128 to this register.

The interrupt routine must read exactly this number of samples out each time it runs. The last time the routine runs, it should read whatever is remaining in the FIFO by monitoring the EF bit (Empty Flag) in the FIFO status register at Base + 7. When the FIFO is empty, EF = 1, and the FIFO returns the value hex FF on all read operations.

If you are sampling at a slow rate or want to control when the interrupt occurs, you can set the threshold to a low value. For example, if you are sampling 16 channels at 10Hz and you want an interrupt each set of samples, you can set the threshold to 16 (write an 8 to this register), so that an interrupt will occur each 16 samples. Then the interrupt routine should read out 16 samples from the FIFO, and you get new data as soon as it is available.

For higher sample rates (100KHz or higher) it may be necessary to increase the threshold above 256, to around 350 or even 512 with enhanced features enabled. If you set the threshold too high, you may overrun the FIFO, since the interrupt routine may not respond before the remaining locations are filled, causing an overflow. An overflow can be detected by checking the OVF bit in the FIFO status register at Base + 7. The correct threshold for your application can only be determined by testing.

Base + 7 Write FIFO Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	FIFOEN	SCANEN	FIFORST	X

FIFOEN FIFO enable:

1 Enable FIFO operation; if interrupts are enabled, interrupts will occur when the FIFO hits threshold (TF = 1). This slows down the interrupt rate dramatically compared to the actual A/D sample rate.

0 Disable FIFO operation; if interrupts are enabled, interrupts will occur after each A/D conversion.

SCANEN Scan enable:

1 Scan mode enabled; FIFO will fill up with data for a single scan, and STS will stay high until entire scan is complete; if interrupts are enabled, interrupts will occur on integral multiples of scans.

0 Scan mode disabled; The STS bit will correspond directly to the status indicator from the A/D converter

FIFORST FIFO reset:

1 Reset FIFO; after this command is issued, EF = 1, TF = 0, FF = 0

0 No function

See the FIFO chapter later in this manual for a complete description of FIFO operation.

Base + 7 Read FIFO Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	EF	TF	FF	OVF	FIFOEN	SCANEN	PAGE1	PAGE0

EF Empty flag:

1 FIFO is empty

0 FIFO is not empty

TF Threshold flag:

1 FIFO is at or beyond threshold; if the FIFO threshold is 256 words, this flag is set when the FIFO contains at least 256 words of A/D data.

0 FIFO is less than threshold

FF Full flag:

1 FIFO is full; the next A/D conversion will result in an overflow

0 FIFO is less than full

OVF Overflow flag:

1 FIFO has overflowed; data has been lost. This flag is cleared on the next successful A/D read.

0 FIFO has not overflowed since the last A/D data read

FIFOEN, SCANEN Read-back of control bits from above

PAGE1-0 Read-back of the current page register setting; see Base + 8 below

Base + 8 Write Miscellaneous Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	RESETA	RESETD	INTRST	**P2**	P1	P0

RESETA Writing a 1 to this bit causes a full reset of all features of the board, including the DACs, the FIFO, the digital I/O, and all internal registers. The counter/timers are not affected by this reset.

RESETD Writing a 1 to this bit causes a reset identical to above except the analog outputs are not affected.

INTRST Writing a 1 to this bit resets the interrupt request circuit on the board. The programmer must write a 1 to this bit during the interrupt service routine, or further interrupts will not occur. Writing a 1 to this bit does not disturb the values of the PAGE bits.

P2-0 Three-bit value that selects which I/O device is accessible through the registers at locations Base + 12 through Base + 15:

P<2:0>	Page	Device
000	0	8254
001	1	8255
010	2	FIFO Control
011	3	EEPROM/TrimDAC
100	4	dsPIC
101	5	D/A Waveform Generator
110	6	Factory use only
111	7	D/A Channel Control

Gray pages (2, 4, 5, 6 and 7) are only accessible when the enhanced features are enabled. Note that P2 is an enhanced feature bit.

Writing to the page bits will not generate a board reset or interrupt reset, as long as those bits are kept at 0 in the data written to this register.

Base + 8 Read A/D Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	STS	S/D1	S/D0	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0

STS A/D chip status:

1 A/D conversion or A/D scan in progress

0 A/D idle

S/D1-0 Single-ended / Differential A/D input mode indicator. S/D1 controls the channels 8-15 and 24-31, S/D0 controls 0-7 and 16-23.

1 Single-ended (default)

0 Differential

ADCH4-0 Current A/D channel; this is the channel currently selected on board and is the channel that will be used for the next A/D conversion (unless a new value is written to the low channel register).

Base + 9 **Write** **Interrupt and A/D Clock Control Register**

Bit No.	7	6	5	4	3	2	1	0
Name	ADINTE	DINTE	TINTE	RSVD1	**DMAEN**	X	CLKEN	CLKSEL

ADINTE A/D interrupt enable:

1 Enable A/D interrupt operation

0 Disable A/D interrupt operation

DINTE Digital interrupt enable:

1 Enable digital I/O interrupt operation.

0 Disable digital I/O interrupt operation

TINTE Timer 0 interrupt enable:

1 Enable counter/timer 0 interrupt operation

0 Disable counter/timer 0 interrupt operation

RSVD1 Reserved for future use

DMAEN DMA Enable. This bit is ignored if enhanced features are disabled. See DMA signal definition for more detail on DMA behavior.

1 DMA Enabled

0 DMA Disabled

CLKEN Enable hardware clock for A/D sampling:

1 Enable hardware clock for A/D (source is selected with CLKSEL bit below);

NOTE: When this bit is 1, software triggers are disabled, i.e. writing to Base + 0 will not start an A/D conversion.

0 Disable hardware clocking for A/D; A/D conversions occur with software command only

CLKSEL Hardware clock select (enabled only when CLKEN = 1 above):

1 Internal clock: **Falling** edges on the output of counter/timer 2 generate A/D conversions. Counter 2 is in turn driven by counter 1, which is driven by the clock selected by bit FREQ12 in Base + 10 below.

0 External trigger: **Falling** edges on the DIN3/EXTCLK pin on the I/O header generate A/D conversions.

Base + 9 **Read** **Interrupt and A/D Clock Status Register**

Bit No.	7	6	5	4	3	2	1	0
Name	ADINT	DINT	TINT	X	**DMAEN**	**P2**	CLKEN	CLKSEL

ADINT A/D interrupt status

1 A/D interrupt request has occured

0 No interrupt request

DINT Digital interrupt status

1 Digital interrupt request has occured

0 No interrupt request

TINT Timer interrupt status

1 Timer interrupt request has occured

0 No interrupt request

DMAEN Read-back of control register bit defined above

P2 Read-back of P2 register bit defined at Base+8/write

CLKEN Read-back of control register bit defined above

CLKSEL Read-back of control register bit defined above

ADINT, DINT and TINT are cleared by writing to INTRST (base+8)

Base + 10 Read/Write Counter and Digital I/O Configuration Register

Bit No.	7	6	5	4	3	2	1	0
Name	FREQ12	FREQ0	OUT2EN	OUT0EN	RSVD	GT0EN	SRC0	GT12EN

FREQ12 Input frequency select for the counter 1-2 cascade:

1 Input to counter 1 is a 100KHz (one hundred, not ten) frequency derived from the on-board 10MHz oscillator

0 Input to counter 1 is 10MHz from the on-board oscillator

FREQ0 Input frequency select for counter 0 when SRC0 = 1 (bit 1):

1 Input to counter 0 is a 10KHz (ten, not one hundred) frequency derived from the on-board 10MHz oscillator

0 Input to counter 0 is 10MHz from the on-board oscillator

OUT2EN Counter/timer 2 output enable:

1 Counter 2 output appears on I/O header J3 pin 42, OUT 2 / DOUT 2

0 OUT 2 / DOUT 2 pin is set by bit DOUT2 at Base + 1

OUT0EN Counter/timer 0 output enable:

1 Counter 0 output appears on I/O header J3 pin 44, OUT 0 / DOUT 0

0 OUT 0 / DOUT 0 pin is set by bit DOUT0 at Base + 1

RSVD Reserved for future use

GT0EN Counter/timer 0 gate enable:

1 Gate 0 / DIN 1, J3 pin 47, acts as an active high gate for counter/timer 0. This pin is connected to a 10KΩ pull-up resistor.

0 Counter/timer 0 runs freely with no gating

SRC0 Counter 0 input source:

1 Input to Counter 0 is the clock determined by FREQ0 (bit 6)

0 Input to Counter 0 is J3 pin 48 (CLK 0 / DIN 0). The falling edge is active. This pin is connected to a 10KΩ pull-up resistor.

GT12EN Counter/timer 1/2 and external trigger gate enable:

This bit enables gating for A/D sampling for both internal and external clocking.

1 When J3 pin 46 (EXTGATE / DIN 2) is low prior to the start of A/D conversions, A/D conversions will not begin until it is brought high (trigger mode).

If the pin is brought low while conversions are occurring, conversions will pause until it is brought high (gate mode).

J3 pin 46 is connected to a 10KΩ pull-up resistor.

0 The interrupt operation begins immediately once it is set up and the selected clock source begins, with no external triggering or gating.

Base + 11 Write Analog Configuration Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	**SCINT1**	**SCINT0**	RANGE	ADBU	G1	G0

This register controls the analog input range for all channels on the board.

SCINT1-0 Scan interval. This is the time between A/D samples when performing a scan (SCANEN = 1). The driver sets a default of 10µs.

Note that a new interval has been added: 4Vs. This interval is valid even if enhanced features are disabled. It has been adjusted from the DMM-32-AT value of 5Vs to allow A/D conversions up to the new higher limit of 250KHz.

SCINT1	SCINT0	Interval
0	0	20µs
0	1	15µs
1	0	10µs
1	1	4µs

RANGE 5V or 10V A/D positive full-scale voltage (0 = 5V, 1 = 10V)

ADBU A/D bipolar / unipolar setting; 0 = bipolar, 1 = unipolar

RANGE	ADBU	A/D Range
0	0	+/-5V
0	1	0-5V
1	0	+/-10V
1	1	0-10V

These two control bits define the A/D input range for a gain setting of 1.

G1 - 0 A/D programmable gain amplifier setting:

G1	G0	Gain
0	0	1
0	1	2
1	0	4
1	1	8

The gain setting is the ratio between the full-scale voltage range at the A/D converter and the full-scale voltage range at the input to the board. The gain should never cause the input signal to exceed the range of the A/D, because incorrect measurements will result (clipping).

On DMM-32DX-AT, the A/D full-scale voltage range is defined by the RANGE and ADBU bits above. To calculate the optimum gain setting, select the highest gain that does not allow the input signal to exceed the selected A/D range over its entire expected fluctuation range. Note that these settings can be changed at any time, even between A/D conversions, so you can tune the board's settings to each input signal.

Note: On power up or system reset, the board is configured for A/D bipolar mode, input range = ±5V, and gain = 1, corresponding to all zeroes in this register.

Base + 11 Read Analog I/O Readback Register

Bit No.	7	6	5	4	3	2	1	0
Name	WAIT	RSVD	**SCINT1**	**SCINT0**	RANGE	ADBU	G1	G0

WAIT Analog input circuit settling time holdoff indicator:

1 The analog input circuit is settling on a new signal and is not yet ready for a new conversion to start; this will occur each time you change the channel, gain, or input range on the board. The wait time is approximately 10μS.

0 The analog input circuit has settled, and a new A/D conversion may begin

SCINT1 Read-back of control bit described above. Not available unless enhanced features are enabled.

SCINT0 Read-back of control bit described above. Not available unless enhanced features are enabled.

RANGE Read-back of control bit described above

ADBU Read-back of control bit described above

G1 Read-back of control bit described above

G0 Read-back of control bit described above

6.4 Page 0: 82C54 Counter/Timer Access

This section is included as a reference to the page 0 counter/timer registers. Behavior of these registers should be identical to the 82C54 counter/timer chip. Please read reference [12], the 82C54 datasheet, for this behavior. More info on counter/timer signals can be found later in this document.

Page 0, Base + 12 Read/Write Counter 0 data

Bit No.	7	6	5	4	3	2	1	0
Name	CTR0D7	CTR0D6	CTR0D5	CTR0D4	CTR0D3	CTR0D2	CTR0D1	CTR0D0

CTR0D7-0 Counter 0 data

Page 0, Base + 13 Read/Write Counter 1 data

Bit No.	7	6	5	4	3	2	1	0
Name	CTR1D7	CTR1D6	CTR1D5	CTR1D4	CTR1D3	CTR1D2	CTR1D1	CTR1D0

CTR1D7-0 Counter 1 data

Page 0, Base + 14 Read/Write Counter 2 data

Bit No.	7	6	5	4	3	2	1	0
Name	CTR2D7	CTR2D6	CTR2D5	CTR2D4	CTR2D3	CTR2D2	CTR2D1	CTR2D0

CTR2D7-0 Counter 2 data

Page 0, Base + 15 Read/Write Counter/timer Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC1-0 Counter Select

RW1-0 Read/Write Mode

M2-0 Timer Mode

BCD Binary Coded Decimal Count

For more information see the 82C54 Datasheet

6.5 Page 1: 82C55 Digital I/O Circuit

This section is included as a reference to the page 1 82C55-like digital I/O registers. More info on the behavior of these digital I/O signals can be found later in this document.

Page 1, Base + 12 Read/Write Digital I/O Port A

Bit No.	7	6	5	4	3	2	1	0
Name	A7	A6	A5	A4	A3	A2	A1	A0

A7-0 Port A data

Page 1, Base + 13 Read/Write Digital I/O Port B

Bit No.	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0

B7-0 Port B data

Page 1, Base + 14 Read/Write Digital I/O Port C

Bit No.	7	6	5	4	3	2	1	0
Name	C7	C6	C5	C4	C3	C2	C1	C0

C7-0 Port C data

Page 1, Base + 15 Read/Write Digital I/O Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	1	ModeC	ModeA	DIRA	DIRCH	ModeB	DIRB	DIRCL

1 Bit 7 must be set to 1. This indicates port configure mode in the 8255 (as opposed to bit set mode which is not supported).

ModeA-C Mode configuration bits. These must be set to 0.

DIRA, DIRB, DIRCH, DIRCL Direction control bits. On ports A and b, all the bits in each port must be the same direction. On port C, the upper half C7 – C4 can have a different direction than the lower half C3 – C0.

0 Output

1 Input

6.6 Page 2: Expanded FIFO Control

This is an enhanced features page. It is inaccessible unless enhanced features are enabled.

Page 1, Base + 12 Read/Write Expanded FIFO Depth Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	X	**FT9**

FT9 This bit is used when setting the FIFO threshold.

See the documentation for Base+6 for more information.

6.7 Page 3: Autocalibration Registers

These registers are used to control the autocalibration process. For user software-controlled autocalibration (as with the DMM-32-AT), these registers are used by the Universal Driver software or the user's software to manage the calibration process. For auto-autocalibration, the onboard dsPIC microcontroller uses these registers to manage the autocalibration automatically.

Page 3, Base + 12 Read/Write EEPROM / TrimDAC Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0

D7-0 Calibration data to be read or written to the EEPROM and/or TrimDAC.

During EEPROM or TrimDAC write operations, the data written to this register will be written to the selected device.

During EEPROM read operations this register contains the data to be read from the EEPROM and is valid after EEBUSY = 0.

The TrimDAC data cannot be read back.

Page 3, Base + 13 Read/Write EEPROM / TrimDAC Address Register

Bit No.	7	6	5	4	3	2	1	0
Name	X	A6	A5	A4	A3	A2	A1	A0

A6-A0 EEPROM / TrimDAC address.

The EEPROM recognizes address 0 – 127 using address bits A6 – A0 of this register. The TrimDAC only recognizes addresses 0 – 7 using bits A2 – A0. In each case remaining address bits will be ignored.

Page 3, Base + 14 Write Calibration Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	EE_EN	EE_RW	RUNCAL	CMUXEN	TDACEN	X	X	X

This register is used to initiate various commands related to autocalibration. More detailed information on autocalibration may be found elsewhere in this manual.

EE_EN EEPROM Enable. Writing a 1 to this bit will initiate a transfer to/from the EEPROM as indicated by the EE_RW bit.

EE_RW Selects read or write operation for the EEPROM: 0 = Write, 1 = Read.

RUNCAL Writing 1 to this bit causes the board to reload the calibration settings from EEPROM.

CMUXEN Calibration multiplexor enable. The cal mux is used to read precision on-board reference voltages that are used in the autocalibration process. It also can be used to read back the value of analog output 0.

1 enable cal mux and disable user analog input channels

0 disable cal mux, enable user inputs

TDACEN TrimDAC Enable. Writing 1 to this bit will initiate a transfer to the TrimDAC (used in the autocalibration process).

Page 3, Base + 14 Read Calibration Status Register

Bit No.	7	6	5	4	3	2	1	0
Name	0	TDBUSY	EEBUSY	CMUXEN	TDACEN	0	0	0

TDBUSY TrimDAC busy indicator

0 User may access TrimDAC

1 TrimDAC is being accessed; user must wait

EEBUSY EEPROM busy indicator

0 User may access EEPROM

1 EEPROM is being accessed; user must wait

CMUXEN Calmux enable status

0 Calibration multiplexor is not currently enabled

1 Calibration multiplexor is enabled and may be updated

TDACEN TrimDAC enable status

0 TrimDAC is not enabled

1 TrimDAC is enabled and may be updated

Page 3, Base + 15 Write Advanced Feature Access Register

After entering page 3 by setting the Page bits, the user must write the value 0xA5 (binary 10100101) to this register in order to get access to the EEPROM. This helps prevent accidental corruption of the EEPROM contents. Once the page is set and this value is written, you can make unlimited reads and writes to the EEPROM without resending this key as long as you stay on page 3.

Writing 0xA6 to this register enables enhanced mode, and writing 0xA7 to this register disables enhanced mode. The default power-on state is disabled.

Enhanced mode consists of the following:

- Pages 2, 4, 5, and 6 are accessible.

Writing 0xA7 to this register disables all enhanced features. This is the default power-on state. Writing 0xA7 to this register automatically halts any enhanced feature currently running, internally clears all enhanced registers to their default state, and resets the A/D FIFO depth to 512.

Page 3, Base + 15 Read FPGA Revision Code

This register indicates the revision level of the FPGA design. This value may change with new revisions of the FPGA, so its value cannot be predicted. It is provided as a way to distinguish between different versions of FPGA code.

6.8 Page 4: dsPIC Interface

This is an enhanced features page. It is inaccessible unless enhanced features are enabled.

Page 4, Base + 12 Read/Write dsPIC Data Register

Bit No.	7	6	5	4	3	2	1	0
Name	PICD7	PICD6	PICD5	PICD4	PICD3	PICD2	PICD1	PICD0

PICD7-0 Data to read/write to/from the PIC microcontroller. The data must be written to this register before the address and read/write bit is written to Base + 13 below.

Page 4, Base + 13 Write dsPIC Address Register

Bit No.	7	6	5	4	3	2	1	0
Name	PICR/W			PICA4	PICA3	PICA2	PICA1	PICA0

PICR/W Read/write control: 0 = write, 1 = read

PICA4-0 dsPIC internal address

Writing a byte with R/W = '0' will cause the dsPIC to write the data contained in the dsPIC Data Register above to the dsPIC internal address indicated by PICA4-0.

Writing a byte with R/W = '1' will cause the dsPIC to read the data at dsPIC internal address, PICA4-0 and place the received data in the dsPIC Data Register.

Page 4, Base + 13 Read dsPIC status register

Bit No.	7	6	5	4	3	2	1	0
Name	I2CBSY			PICA4	PICA3	PICA2	PICA1	PICA0

I2CBSY I2C port status bit:

0 Last I2C operation completed

1 Last I2C operation in progress

PICA4-0 dsPIC address last accessed

Page 4, Base + 14 Write Auto-Autocalibration Command Register

Bit No.	7	6	5	4	3	2	1	0
Name				ACHOLD	ACREL	PICRST	ACABT	ACTRIG

Only one bit can be set to '1' at once. Bits are processed MSB to LSB. The first '1' determines which command is carried out.

ACHOLD 1 = Auto-autocal process is disabled. Autocalibration must be triggered by software

ACREL 1 = Auto-autocal process is enabled. Auto-autocalibration will occur whenever the board requires it.

PICRST 1 = Reset dsPIC device. This command is normally not needed.

ACABT 1 = Abort any currently running auto-autocal operation immediately

ACTRIG 1 = Initiate an auto-autocal process immediately

Page 4, Base + 14 Read Auto-Autocalibration Status Register

Bit No.	7	6	5	4	3	2	1	0
Name				ACHOLD	PICPRST	ACERR	ACACT	PICBSY

ACHOLD 1 = dsPIC in holdoff mode (auto-autocal disabled)

PICPRST 1 = dsPIC device present on board

ACERR 1 = dsPIC detected errors during last Auto-autocal process

ACACT 1 = Auto-autocal process currently in progress

PICBSY 1 = dsPIC busy, either with auto-autocal or some other activity

Details concerning auto-autocalibration can be found at in chapter 15 on page 48.

Page 4, Base + 15 Write dsPIC Programming Register

Bit No.	7	6	5	4	3	2	1	0
Name	PSTART	PSTOP	PGDOUT	PGDIN	PGDW1	PGDW0	PGCW1	PGCW0

This register is used to control the on-board dsPIC microcontroller. The dsPIC controls the auto-autocalibration process, and it also provides the communication link between the board and its serial port.

The bits in this register are control bits, not register bits. Only one bit can be set to '1' at once. Bits are processed MSB to LSB. The first '1' determines which command is carried out.

PSTART Drive EN_PROG# signal low.

PSTOP Drive EN_PROG# high.

PGDOUT FPGA makes PIC_PGD line an output, but leave at current level (i.e. perform input to find current level, set line as an output at same level.)

PGDIN FPGA makes PIC_PGD line an input.

PGDW1 If PIC_PGD line is in output mode, set high

PGDW0 If PIC_PGD line is in output mode, set low

PGCW1 Set PIC_PGC line high

PGCW0 Set PIC_PGC line low

Page 4, Base + 15 Read dsPIC Programming Register

Bit No.	7	6	5	4	3	2	1	0
Name								PGDR

PGDR Reads back current level of PIC_PGD line (low = 0, high = 1)

6.9 Page 5: D/A Waveform Generator

This is an enhanced features page. It is inaccessible unless enhanced features are enabled.

Page 5, Base + 12 Write Store D/A Code at Address (LSB)

Bit No.	7	6	5	4	3	2	1	0
Name	DACA7	DACA6	DACA5	DACA4	DACA3	DACA2	DACA1	DACA0

DACA7-0 LSB of address to store D/A code in D/A waveform buffer

Page 5, Base + 13 Write Store D/A Code at Address (MSB)

Bit No.	7	6	5	4	3	2	1	0
Name							DACA9	DACA8

DACA9-8 MSB of address to store D/A code in D/A waveform buffer

Page 5, Base + 14 Read/Write Waveform Generator Control Register

Bit No.	7	6	5	4	3	2	1	0
Name	DEPTH3	DEPTH2	DEPTH1	DEPTH0	WGCH1	WGCH0	WGSRC1	WGSRC0

DEPTH3-0 These bits define the size of the D/A waveform buffer. The depth is based on this equation:

Depth = [(DEPTH3-0) + 1] * 64

This allows valid depth values from 64 to 1024 samples.

The waveform generator frame pointer will return to 0 whenever it hits either 1024 or the depth value indicated above.

WGCH1-0 These two bits combine to choose how many codes are output on each frame.

WGCH 1	WGCH 0	Description
0	0	1 code per frame
0	1	2 codes per frame
1	X	4 codes per frame

WGSRC1-0 These two bits combine to choose which trigger source is used to increment the waveform by one frame

WGSRC 1	WGSRC0	Description
0	0	Manual (using WGINC)
0	1	Counter 0 output
1	0	Counter 1/2 output
1	1	External trigger (J3, pin 45)

Page 5, Base + 15 Write Waveform Generator Command Register

Bit No.	7	6	5	4	3	2	1	0
Name					WGINC	WGRST	WGPS	WGSTRT

Only one bit can be set to '1' at once. Bits are processed MSB to LSB. The first '1' determines which command is carried out.

WGSTRT Begin or resume the waveform generator

WGPS Pause/stop the waveform generator. The current position in memory is saved for the next begin/resume, or can be reset using WGRST.

WGRST Reset the waveform generator to output from the beginning of the D/A code buffer

WGINC Force the waveform generator to increment one frame

Details concerning D/A waveform generator can be found in Chapter 14 on page 46.

6.10 Page 6: CPLD I/O Window

The CPLD I/O is an internal device and should not be used under standard operation. If more information on this device is required, please contact Diamond Systems.

6.11 Page 7: D/A Output Channel Control

Page 7, Base + 12 Write D/A Output Low Byte

Bit No.	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	D/A BIT 07	D/A BIT 06	D/A BIT 05	D/A BIT 04	D/A BIT 03	D/A BIT 02	D/A BIT 01	D/A BIT 00

D/A BIT 07-00 LSB of Data sent to the D/A converter.

Page 7, Base + 13 Write D/A Output High Byte

Bit No.	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	D/A BIT 15	D/A BIT 14	D/A BIT 13	D/A BIT 12	D/A BIT 11	D/A BIT 10	D/A BIT 09	D/A BIT 08

D/A BIT 15-08 MSB of Data sent to the D/A converter.

These registers are used to write 16-bit data to the D/A. These registers function correctly regardless of whether the installed D/A is 12 bits or 16 bits. If data is written to page 7 base + 13, then the next time base + 5 is written, the FPGA will send the data from page 7 base + 12 & 13 to the channel selected by the channel bits in base + 5. After one such cycle, the FPGA will revert to using base + 4 & 5 for the D/A data until page 7 base + 13 is written to again.

Page 7, Base + 14 Write D/A Output Low Byte

Bit No.	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				D/A JPOVRD	D/A FX/VR	D/A GAIN1	D/A GAIN0	D/A POL

D/A JPOVRD Selects control of the D/A converter between the hardware jumper settings of signals J_D/A FX/VR, J_D/A GAIN1, J_D/A GAIN0 and J_D/A POL or the register bits D/A FX/VR, D/A GAIN1, D/A GAIN0 and D/A POL here in Register 14.

When set to '1' use the contents of this register
When set to '0' use the jumpers on board.

D/A FX/VR Sets the D/A converter reference voltage to variable when '0' or fixed when '1'.

D/A GAIN1 Sets the D/A converter output voltage gain. See table below.

D/A GAIN0 Sets the D/A converter output voltage gain. See table below.

D/A POL Sets the D/A converter polarity to unipolar when '0' or bipolar when '1'. See table below.

D/A GAIN1	D/A GAIN0	D/A POL	Description
0	0	0	5V span (0 to +5V unipolar)
0	0	1	5V span (-2.5V to +2.5V, biplolar)
0	1	0	10V span (0 to +10V unipolar)
0	1	1	10V span (-5V to 5V bipolar)
1	0	0	Not Used – Sets output to 0V
1	0	1	20V span (-10V to +10V only)
1	1	0	Not Used – Sets output to 0V
1	1	1	D/A converter shut down

Page 7, Base + 14 Read D/A Hardware Jumper Configuration

Bit No.	7	6	5	4	3	2	1	0
Name		**J_DAC SZ1**		**D/A JPOVRD**	**D/A FX/VR**	**D/A GAIN1**	**D/A GAIN0**	**D/A POL**

J_DAC SZ1 Indicates the D/A converter size as indicated in the table below:

J_DAC SZ1	Description
0	16 bits
1	12 bits

D/A JPOVRD Indicates status of the control of the D/A converter between the hardware jumper settings of signals J_D/A FX/VR, J_D/A GAIN1, J_D/A GAIN0 and J_D/A POL or the register bits D/A FX/VR, D/A GAIN1, D/A GAIN0 and D/A POL here in Register 14.

When set to '1' use the write contents of this register
When set to '0' use the jumpers on the board.

D/A FX/VR Set the D/A converter reference voltage to variable when '0' or fixed when '1'.

D/A GAIN1 Indicates the D/A converter output voltage gain setting. See table above.

D/A GAIN0 Indicates the D/A converter output voltage gain setting. See table above.

D/A POL Indicates the D/A converter polarity is set to unipolar when '0' or bipolar when '1'. See table above.

7. ENABLING ENHANCED FEATURES

The DMM-32DX-AT has many newly added features that are only accessible when “enhanced features” is enabled. These features include D/A wave form generator (access to Page 5), larger FIFO size of 1024 (access to Page 2), dsPIC and auto autocalibration (access to Page 4), and various others.

7.1 Enabling Enhanced Features (Enhanced Mode)

Two steps are required to enable enhanced features:

1. Set page bits to Page 3
2. Write code to unlock enhanced features

Below is the code demonstrating how to enable enhanced features without using the driver software,

The page bit can be set at Base + 8, (see page 21):

```
outp(Base + 8, 0x3);
```

Write 0xA6 to Base + 15 to unlock enhanced features (see page 28):

```
outp(Base + 15, 0xA6);
```

7.2 Disabling Enhanced Features (Normal Mode)

To disable enhanced features follow the same instructions as above except write 0xA7 instead of 0xA6 to Base + 15.

```
outp(Base +15, 0xA7);
```

8. ANALOG INPUT RANGES AND RESOLUTION

Diamond-MM-32DX-AT uses a 16-bit A/D converter. This means that the analog input voltage can be measured to the precision of a 16-bit binary number. The maximum value of a 16-bit binary number is 2^{16} - 1, or 65535, so the full range of numerical values that you can get from a Diamond-MM-32DX-AT analog input channel is 0 - 65535.

The smallest change in input voltage that can be detected is $1/(2^{16})$, or 1/65536, of the full-scale input range. This smallest change results in an increase or decrease of 1 in the A/D code, and so this change is referred to as 1 LSB, or 1 least significant bit.

8.1 Unipolar and Bipolar Inputs

Diamond-MM-32DX-AT can measure both unipolar (positive only) and bipolar (positive and negative) analog voltages. The full-scale input voltage range depends on the Gain, Range, and Polarity bit settings in the Analog Configuration Register at Base + 11. In front of the A/D converter is a programmable gain amplifier that multiplies the input signal before it reaches the A/D. This gain circuit has the effect of scaling the input voltage range to match the A/D converter for better resolution. In general you should select the highest gain you can that will allow the A/D converter to read the full range of voltages over which your input signals will vary. If you pick too high a gain, then the A/D converter will clip at either the high end or low end, and you will not be able to read the full range of voltages on your input signals.

8.2 Input Ranges and Resolution

The table below lists the full-scale input range for each valid analog input configuration. The parameters Polarity, Range, and Gain are combined to create the value "Code", which is the value that you must write to the analog configuration register at Base + 11 to get the input range shown. A total of 9 different input ranges are possible. Note that the range programming codes 4, 5, 6, and 7 are invalid and that range codes 9 – 11 are equivalent to range codes 0 – 2.

Diamond-MM-32DX-AT Analog Input Ranges

Polarity	Range	Gain	Code	Full-Scale Range	Resolution (1 LSB)
Bipolar	5V	1	**0**	±5V	153 μV
Bipolar	5V	2	**1**	±2.5V	76 μV
Bipolar	5V	4	**2**	±1.25V	38 μV
Bipolar	5V	8	**3**	±0.625V	19 μV
Unipolar	5V	1	**4**	Invalid setting	
Unipolar	5V	2	**5**	Invalid setting	
Unipolar	5V	4	**6**	Invalid setting	
Unipolar	5V	8	**7**	Invalid setting	
Bipolar	10V	1	**8**	±10V	305 μV
Bipolar	10V	2	**9**	±5V	153 μV
Bipolar	10V	4	**10**	±2.5V	76 μV
Bipolar	10V	8	**11**	±1.25V	38 μV
Unipolar	10V	1	**12**	0 - 10V	153 μV
Unipolar	10V	2	**13**	0 - 5V	76 μV
Unipolar	10V	4	**14**	0 - 2.5V	38 μV
Unipolar	10V	8	**15**	0 - 1.25V	19 μV

8.3 A/D Conversion Formulas

The 16-bit value returned by the A/D converter is always a twos complement number ranging from -32768 to 32767, regardless of the input range. This is because the input range of the A/D is fixed at ±10V. The input signal is actually magnified and shifted to match this range before it reaches the A/D. For example, for an input range of 0-10V, the signal is first shifted down by 5V to ±5V and then amplified by 2 to become ±10V.

Therefore, two different formulas are needed to convert the A/D value back to a voltage, one for bipolar ranges, and one for unipolar ranges. Tables showing the correlation between A/D code and input voltage are shown on the following page.

For Bipolar Input Ranges

FS = full-scale voltage (e.g. 5V for ±5V range)

If using a 16-bit signed integer in C:

Input voltage = (A/D code / 32768) x FS

Example: ±5V range selected, A/D code = **17762** (Hex 4560)

Input voltage = (17762 / 32768) x 5V = **+2.7103V**

Example: ±5V range selected, A/D code = **-15008** (Hex C560)

Input voltage = (-15008 / 32768) x 5V = **-2.2900V**

If using a 32-bit signed integer in C, or unsigned or floating value in C or Basic:

Input voltage = (A/D code / 32768) x FS

If input voltage >= FS then input voltage = input voltage - 2 x FS

Example: ±5V range selected, A/D code = **17762** (Hex 4560)

Input voltage = (17762 / 32768) x 5V = **+2.7103V**

Example: ±5V range selected, A/D code = **50528** (Hex C560)

Input voltage = (50528 / 32768) x 5V = +7.7100V

Since 7.7100V >= 5V, we must subtract:

Input voltage = 7.7100V - 2 x 5V = **-2.2900V**

For Unipolar Input Ranges

FS = full-scale voltage (e.g. 10 for 0 - 10V range)

Input voltage = ((A/D code + 32768) / 65536) x FS

Example: 0 - 10V range selected, A/D code = **17762** (Hex 4560)

Input voltage = ((17762 + 32768) / 65536) x 10V = **+7.7103V**

Note that this is simply the result for the ±5V range shifted up by 5V.

8.4 Correlation between A/D Code and Input Voltage

The two tables below illustrate the correlation between the A/D code and the corresponding input voltage. Use these tables as guides to help think about how to convert between the voltage domain and the A/D code domain.

Bipolar Input Ranges

A/D Code	Input voltage formula	Input voltage (±5V range)
-32768	$-V_{FS}$	-5.0000V
-32767	$-V_{FS}$ + 1 LSB	-4.9998V
-1	-1 LSB	-0.153mV
0	0V	0.0000V
1	+1 LSB	0.153mV
32767	V_{FS} - 1 LSB	4.9998V

Unipolar Input Ranges

A/D Code	Input voltage formula	Input voltage (0 - 10V range)
-32768	0V	0.0000V
-32767	1 LSB (V_{FS} / 65536)	0.153mV
-1	V_{FS} / 2 - 1 LSB	4.99985V
0	V_{FS} / 2	5.0000V
1	V_{FS} / 2 + 1 LSB	5.00015V
32767	V_{FS} - 1 LSB	9.9998V

9. PERFORMING AN A/D CONVERSION

This chapter describes the steps involved in performing an A/D conversion on a selected input channel using direct programming (not with the driver software).

The A/D FIFO

All A/D conversions are stored in an on-board FIFO (first in first out memory). The FIFO can hold up to 1024 samples. Each time an A/D conversion is finished, the data is stored in the FIFO, and the FIFO counter increments by 1. Each time you read A/D data, you are actually reading it out of the FIFO, and the FIFO counter decrements by 1. When the FIFO is empty the data read from it is undefined – you may continue to read the last sample, or you may read all 1s.

You can read each A/D sample as soon as it is ready, or you can wait until you take a collection of samples (up to 1024 maximum) and then read them all out at once.

To be sure that you are getting only current A/D data, be sure to reset the FIFO each time before you start any A/D operation. This will prevent errors caused by leaving data in from a previous operation. To reset the FIFO, write a 1 to bit 2 of register 7 (see page 20). This bit is not a real register bit; instead it triggers a command in the board's controller chip. Therefore you do not need to write a 1 and then a 0, just write a 1. However writing to the FIFORST bit affects the values of other bits in this register as well:

```
outp(Base+7, 0x02); // resets the FIFO and clears SCANEN and FIFOEN
outp(Base+7, 0x0A); // resets the FIFO and SCANEN but leaves FIFOEN set
```

Note that this register also contains a FIFO enable bit, FIFOEN. This bit only has meaning during A/D interrupt operations. The FIFO is always enabled and is always in use during A/D conversions.

There are seven steps involved in performing an A/D conversion:

1. **Select the input channel or input channel range**
2. **Select the analog input range (Range, Polarity, and Gain codes)**
3. **Wait for analog input circuit to settle**
4. **Start an A/D conversion on the current channel**
5. **Wait for the conversion to finish**
6. **Read the A/D data**
7. **Convert the numerical data to a meaningful value**

If you are going to sample the same channel multiple times or sample multiple consecutive channels with the same input range, you only need to perform steps 1-3 once, and then you can repeat steps 4-6 or 4-7 as many times as desired.

Diamond Systems also provides sample register level code, downloadable at http://diamondsystems.com/files/binaries/SourceCodeExamples.zip

9.1 Select the input channel or input channel range

Diamond-MM-32DX-AT contains a channel counter circuit that controls which channel will be sampled on each A/D conversion command. The circuit uses two channel numbers called the low channel and high channel. These are stored in registers at Base + 2 and Base + 3 (see page 17). The circuit starts at the low channel and automatically increments after each A/D conversion until the high channel is reached. When an A/D conversion is performed on the high channel, the circuit resets to the low channel and starts over again. This behavior enables you simplify your software by setting the channel range just once.

To read continuously from a single channel, write the same channel number to both the low channel and high channel registers.

To read from a series of consecutively numbered channels, write the starting channel to Base + 2 and the ending channel to Base + 3.

To read from a group of non-consecutive channels, you must treat each as a single channel described above.

9.2 Select the analog input range

Select the code from page 35 corresponding to the desired input range and write it to the analog I/O control register at Base + 11. You only need to write to this register if you want to select a different input range from the one used for the previous conversion. If all channels will be using the same input range, you can configure this register just once at the beginning of your procedure.

You can read the current value of this register by reading from Base + 11.

9.3 Wait for the analog circuit to settle

After changing either the input channel or the input range, you must allow the circuit to settle on the new value before performing an A/D conversion. The settling time is long compared to software execution times, so a timer is provided on board to indicate when it is safe to proceed with A/D sampling. The WAIT bit at Base + 11 (see page 24) indicates when the circuit is settling and when it is safe to sample the input. When WAIT is 1 the board is settling; when WAIT is 0 the board is ready for an A/D conversion.

9.4 Start an A/D conversion on the current channel

To generate an A/D conversion, simply write to Base + 0 to start the conversion. The value you write does not matter and is ignored.

9.5 Wait for the conversion to finish

The A/D converter takes about 4 microseconds to complete a conversion. If you try to read the A/D converter data immediately after starting a conversion, you will get invalid data. Therefore the A/D converter provides a status signal to indicate whether it is busy or idle. This signal can be read back as the STS bit in the status register at Base + 8 (see page 21). When the A/D converter is busy (performing an A/D conversion), this bit is 1, and when the A/D converter is idle (conversion is done and data is available), this bit is 0.

9.6 Read the A/D data

Once the conversion is complete, you can read the data back from the A/D converter in two 8-bit bytes at Base + 0 and Base + 1 (see page 17). The low byte must be read first. The following pseudocode illustrates how to construct the 16-bit A/D value from these two bytes:

```
LSB = read(Base)          ;Get low 8 bits first
MSB = read(Base+1)        ;Get high 8 bits last
Data = MSB * 256 + LSB    ;Combine the 2 bytes into a 16-bit value
```

The final data ranges from 0 to 65535 (0 to 2^{16} - 1) as an unsigned integer. This value must be interpreted as a signed integer ranging from -32768 to +32767.

As noted above, all A/D conversions are stored in an on-board FIFO, which can hold up to 1024 samples in enhanced mode or 512 samples in normal mode. Whenever you read A/D data you are actually reading it out of the FIFO. Therefore, you can read each A/D sample as soon as it is ready, or you can wait until you take a collection of samples (up to 1024 maximum) and then read them all out in sequence.

9.7 Convert the numerical data to a meaningful value

Once you have the A/D value, you need to convert it to a meaningful value. The formulas on page 36 show you how to convert the A/D data back to the original input voltage. You may want to convert it into engineering units afterwards or instead. The two conversions can be done sequentially, or the formulas can be combined into a single formula.

10. A/D SAMPLING METHODS

10.1 Sampling Modes

There are several different A/D sampling modes available on Diamond-MM-32DX-AT. The mode in use is selected with the FIFO enable and Scan enable bits at the FIFO control register at Base + 7 as well as the A/D interrupt enable bit in the Interrupt control register at Base + 9.

Note that the FIFO should not be enabled if interrupts are not enabled, as the FIFO storage is only useful when interrupts are being used and will have no effect otherwise.

All these features may be selected as arguments to function calls in the driver software. The control register details are provided for completeness and for programmers not using the driver.

SCAN	FIFO	Interrupt	Mode	Description
No	No	No	Single conversions	The most basic sampling method. Used for low-speed sampling (typically up to about 100Hz) under software control where a precise rate is not required, or under external control where the rate is slow. Consists of either one channel or multiple channels sampled one at a time.
Yes	No	No	Scan conversions	Used to sample a group of consecutively numbered channels in rapid succession, under software or external control. The time between samples in a scan is programmable between 5 to 20 microseconds, while the time between scans depends on the software or external trigger and may be very short or very long, but is usually less than about 100Hz (above this rate use interrupt scans below).
No	No	Yes	Interrupt single conversions, low speed	Used for controlled-rate sampling of single channels or multiple channels in round-robin fashion, where the frequency of sampling must be precise but is relatively slow (less than 100Hz). The sampling clock comes from the on-board counter/timer or from an external signal. The interval between all A/D samples is identical.
Yes	No	Yes	Interrupt scans, low speed	Used for controlled-rate sampling a group of channels in low-speed mode (less than 500Hz per channel). Each sampling event consists of a group of channels sampled in rapid succession. The time between scans is determined by the sample rate.
No	Yes	Yes	Interrupt single conversions, how speed	Intended for medium- to high-speed operation (recommended above about 500Hz). Can support sampling rates up to the board's maximum of 250,000Hz. May also be used at slower rates if desired. The sampling clock comes from the on-board counter/timer or from an external signal.
Yes	Yes	Yes	Interrupt scan conversions	Used for high-speed sampling of a group of channels where the scan rate is high. The sampling clock comes from the on-board counter/timer or from an external signal.

10.2 FIFO Description

Diamond-MM-32DX-AT uses a 1024-sample FIFO (First In First Out) memory buffer to manage A/D conversion data. It is used to store A/D data between the time it is generated by the A/D converter and the time it is read by the user program. In enhanced mode the entire 1024-sample FIFO is available, and in normal mode only 512 samples are available. The FIFO may be enabled and disabled under software control.

In single-conversion mode, the FIFO features are not generally needed so FIFO use should not be selected. However the FIFO is still actually being used. Each A/D sample is stored in the FIFO, and when the software reads the data, it reads it out of the FIFO. In low-speed sampling, each time a conversion occurs, the program reads the data, so there is always a one to one correspondence between sampling and reading. Thus the FIFO contents never exceed one sample.

For high-speed sampling or interrupt operation, the FIFO substantially reduces the amount of software overhead in responding to A/D conversions as well as the interrupt rate on the bus, since it enables the program to read a number of samples all at once rather than one at a time. In addition, the FIFO is required for sampling rates in excess of the maximum interrupt rate possible on the bus. Generally, the fastest sustainable interrupt rate on the ISA bus running DOS is around 40,000 per second. Since Diamond-MM-32DX-AT can sample up to 250,000 times per second, the FIFO is needed to reduce the interrupt rate at high speeds. When the interrupt routine runs, it reads multiple samples from the FIFO. The interrupt rate is equal to the sample rate divided by the number of samples read each interrupt. On Diamond-MM-32DX-AT, this number is programmable using the register at Base + 6. The usual value is 1/2 the maximum FIFO depth, or 512 samples. Therefore, the maximum interrupt rate for Diamond-MM-32DX-AT is reduced to 488 per second, easily sustainable on any popular operating system.

IMPORTANT NOTE: If both Scan and FIFO operation are enabled, then the interrupt will still occur at the programmed FIFO threshold, and the interrupt routine will read the indicated number or samples and then exit. This will happen even if the number of samples is not an integral number of scans. For example, if you have a scan size of 10 and a FIFO threshold of 256, then the first time the interrupt routine runs, it will read 256 samples, consisting of 25 full scans of all 10 channels and then 6 samples from the next scan. The next time the interrupt routine runs, it will read the next 256 samples, consisting of the remaining 4 samples from the last scan it started to read, the next 25 full scans of 10 samples, and then the first 2 samples of the next scan. This continues until the interrupt routine ends in either one-shot or recycle mode. In one-shot mode, the last time the interrupt routine runs it will read the entire contents of the FIFO, so that all data will be made available.

10.3 Scan Sampling

A scan is defined as a quick burst of samples of multiple consecutive channels. For example, you may want to sample channels 0-15 all at once, and repeat the operation each second. This would be a scan at a frequency of 1Hz. Each time the A/D clock occurs (software command, timer, or external trigger), all 16 channels are sampled in high-speed succession. There is a short delay of 4 – 20 microseconds between each sample in the scan. Since each clock pulse causes all channels to be sampled, the effective sampling rate for each channel is the same as the programmed rate, and the total sampling rate is the programmed sampling rate times the number of channels in the scan range.

Scan sampling is independent of FIFO operation. Either or both can be enabled independently.

10.4 Sequential Sampling

In sequential sampling, each clock pulse results in a single A/D conversion on the current channel. If the channel range is set to a single channel (high channel = low channel), each conversion is performed on the same input channel. If the channel range is set to more than one channel (high channel > low channel), then the channel counter increments to the next channel in the range, and the next conversion is performed on that channel. When a conversion is performed on the high channel, the channel counter resets to the low channel for the next conversion. The intervals between all samples are equal. Since each clock pulse results in only one channel being sampled, the effective sampling rate is the programmed sampling rate divided by the number of channels in the channel range.

11. HOW TO PERFORM A/D CONVERSIONS USING INTERRUPTS

Diamond-MM-32 contains the ability to generate hardware interrupts to manage A/D conversions. Interrupt-based A/D conversions are used in several situations:

- High-speed sampling
- Applications where the sampling rate must be precise
- Applications where the sampling rate is based on an external clock

The Diamond Systems Universal Driver functions **dscADSampleInt()** and **dscADSetSettings()** manage all of the required parameters to generate interrupt-based A/D conversions. Below is a checklist to help you configure the function call properly. All parameters are passed in the data structure of type **DSCAIOINT** for function **dscADSampleInt()** except for the input range.

1. A/D channel range (low channel, high channel)

On Diamond-MM-32DX-AT, the channel numbers range from 0 to 31. Some channel numbers may not be available, depending on the single-ended / differential configuration mode as explained on page 11. During interrupt-based A/D conversions, the channels being sampled must be consecutive in number. To sample only a single channel, set the low channel and high channel to the same channel number. To sample a range of channels, set the low and high channels accordingly.

2. Input voltage range

During interrupt-based A/D conversions, the input voltage range must be the same for all channels. Select the input range from the list of codes on page 35. This parameter is set with the function **dscADSetSettings()** prior to calling **dscADSampleInt()**.

3. A/D Clock source, internal or external

For internal clocking, the on-board 32-bit counter/timer is programmed to the desired sample rate. For external clocking, the signal EXTCLK / DIN3 on I/O header J3 pin 45 controls sampling. Falling edges on this pin will generate A/D conversions. The signal is edge sensitive, so holding it low will generate only one conversion.

4. A/D conversion rate, if using internal clock

If internal clocking is selected, provide the desired sample rate in Hz as a floating value. The maximum sample rate is 250,000 per second (maximum A/D operating speed), and the slowest rate is .000024383Hz (100KHz input / 232), or approximately 1 sample every 42,950 seconds (approximately 11.9 hours).

5. External gating enable

You can choose whether to allow an external signal on J3 pin 46 to control the sampling. If so, then when this signal is high, sampling will occur, and when it is low, sampling will pause. External gating works with both internal and external clocking. This pin is connected to a 4.7K pull-up resistor.

6. One-shot vs. recycle mode

In one-shot mode, the operation occurs one time and then stops, and the parameter num_conversions determines the number of samples taken. In recycle mode, the operation runs repeatedly until you stop the operation with **dscCancelOp()**. In this case, the parameter num_conversions indicates the size of the memory buffer or array used to store the samples. Once the buffer is filled, the data is stored starting at the beginning again, causing the old data to be overwritten. In this situation, you only have access to the latest number of samples equal to num_conversions, and you must read the data out of the buffer before it is overwritten. The function **dscGetStatus()** can be used to indicate the current buffer position, which is the location at which the next data value will be stored.

12. ANALOG OUTPUT RANGES AND RESOLUTION

12.1 Description

Diamond-MM-32DX-AT uses a four-channel 16-bit standard (12-bit optional) D/A converter (DAC) to provide four analog outputs. A 16-bit DAC can generate output voltages with the precision of a 16-bit binary number. The maximum value of a 16-bit binary number is 2^{16} - 1, or 65535, so the full range of numerical values that you can write to the analog outputs on Diamond-MM-32DX-AT is 0 – 65535 (or 0 – 4095).

⇒ **Note:** In this manual, the terms analog output, D/A, and DAC are all used interchangeably to mean the same thing.

12.2 Resolution

The *resolution* is the smallest possible change in output voltage. For a 16-bit DAC the resolution is 1/(2^{16}), or 1/65536, of the full-scale output range (or 1/(2^^{12}), or 1/4096 of full-scale for optional 12-bit D/A) . This smallest change results from an increase or decrease of 1 in the D/A code, and so this change is referred to as 1 LSB, or 1 least significant bit. The value of this LSB is calculated as follows:

1 LSB = Maximum voltage swing / 65536 (16-Bit)
= Maximum voltage swing / 4096 (12-Bit)

The maximum voltage swing is defined as the difference between the highest nominal output voltage and the lowest output voltage. For an output range of 0-10V or +/-5V, the maximum voltage swing is 10V.

Example:

Output range = +/-5V

Maximum voltage swing = 10V

1 LSB = 10V / 65536 = 152.6uV (16-Bit)
= 10V / 4096 = 2.44mV (12-Bit)

12.3 Full-Scale Range Selection

The D/A converter chip on Diamond-MM-32DX-AT requires two references, one for the low end and one for the high end of the range. The high end can be set to 5V, 10V, or Programmable, and the low end can be either 0V (for unipolar output ranges) or minus the high-end voltage. See page 12 for information on configuring the D/A range. All channels are set to the same output range.

On power up, the D/A automatically resets to the range and polarity set by the hardware jumpers and with the output voltage set to 0V.

13. GENERATING AN ANALOG OUTPUT

This chapter describes the steps involved in generating an analog output (also called performing a D/A conversion) on a selected output channel using direct programming (not with the driver software).

There are six steps involved in performing a D/A conversion:

1. **Compute the D/A output value for the desired output voltage**
2. **Compute the LSB and MSB values**
3. **Add the channel number to the MSB**
4. **Set D/A Simultaneous Update bit**
5. **Write the LSB and MSB to the board**
6. **Monitor the DACBUSY status bit**

13.1 Compute the D/A code for the desired output voltage

A different formula is required for bipolar and unipolar output ranges.

Unipolar Mode D/A Formula

Output value = (Output voltage) / (Full-scale voltage) * (65536 Standard, 4096 Optional)

Example: Desired output voltage = 2.168V, full-scale voltage = 5V, unipolar mode (0-5V)

Output code = 2.168V / 5V * 65536 = 28416 (16-Bit)
= 2.168V / 5V * 4096 = 1776 (12-Bit)

Bipolar Mode D/A Formula

Output value = (Output voltage) / (Full-scale voltage) * 32768 + 32768 (16-Bit)
= (Output voltage) / (Full-scale voltage) * 2048 + 2048 (12-Bit)

Example: Desired output voltage = -2.168V, full-scale voltage = 5V, bipolar mode (±5V)

Output code = -2.168V / 5V * 32768 + 32768 = 18560 (16-Bit)
= -2.168V / 5V * 2048 + 2048 = 1160 (12-bit)

⇒ **Note:** The DAC cannot generate the actual full-scale reference voltage; to do so would require an output code of 65536 (or 4096 with 12-bit option), which is not possible with a 16-bit (or 12-bit) number. The maximum output value is 65535 (or 4095). Therefore the maximum possible output voltage is 1 LSB less than the full-scale reference voltage.

13.2 Compute the LSB and MSB values

Use the following formulas to compute the LSB and MSB values:

LSB = (D/A Code) AND 255 ;keep only the low 8 bits

MSB = int((D/A code) / 256) ;strip off low 8 bits, keep 4 high bits

Example: Output code = 1776
LSB = 1776 AND 255 = 240 (F0 Hex)
MSB = int(1776 / 256) = int(6.9375) = 6

(In other words, 1776 = 6*256 + 240)

13.3 Add the channel number to the MSB

The channel no. is 0-3. It must be inserted in bits 7-6 of the D/A MSB byte written to Base + 5 (see page 18). Here is an example of how to do it:

MSB = MSB + Channel * 64

13.4 Set D/A Simultaneous Update bit

To update the DAC, set DASIM bit to 0. This will perform an update of the current channel and all previously latched channels, causing a simultaneous update. If no other channels were previously latched, then this will only update the current channel. To latch channel set DASIM bit to 1.

To update: MSB = MSB & 0xDF

To latch: MSB = MSB + 32

13.5 Write the LSB and MSB to the board

The LSB is written to Base + 4, and the MSB/channel no. is written to Base + 5. If using enhanced features make sure to enable enhanced features before writing to Base + 4 and Base + 5.

13.6 Monitor the DACBUSY status bit

DACBUSY = 1 for 10uS while the data in registers 4 and 5 are serially shifted into the D/A chip. After DACBUSY returns to 0 users can write to register 4 and 5. Registers 4 and 5 should NOT be written to while DACBUSY = 1. If updating multiple channels for simultaneous update repeat steps 1 to 6.

Example

D/A is set to +/-5V range. Set channel 1 to 3.000V.

1. Compute D/A code

   Using the bipolar mode formula, we compute D/A code = 3V / 5V * 2048 + 2048 = 3276.8.
   Round this up to 3277. (Binary value = 1100 1100 1101)

2. Compute LSB and MSB.

   LSB = 3277 & 255 = 205 (Binary value = 1100 1101)

   MSB = int(3277/256) = 12 (Binary value = 1100)

3. Add channel number to MSB

   MSB = 12 + 1 * 64 = 76

4. Check DASIM. For non-simultaneous update DASIM = 0, for latching DASIM = 1.

   To update: MSB = MSB & 0xDF

   To latch: MSB = MSB + 32

5. Write LSB and MSB to board, enable enhanced features if using latching.

   ```
   outp(Base + 8, 3);          //this is for enabling enhanced features. Set page bit to 3
   outp(Base + 15, 0xA6);      //enable enhanced features
   outp(Base + 4, LSB);
   outp(Base + 5, MSB);
   ```

6. Monitor DACBUSY bit, Base + 4 bit 7

   ```
   while (inp(Base + 4) & 0x80);
   ```

14. D/A WAVEFORM GENERATOR

14.1 Description

Page 5 of the upper I/O map provides control for the D/A waveform generator. The D/A waveform generator uses an in-FPGA memory block of 1024 words to store D/A codes. The FPGA parses through this memory at a user-programmable speed (or through manual/external trigger) while sending codes to the D/A converter. The generator automatically stops if enhanced features are disabled.

The generator works in frames. A new frame is triggered from a programmable source (manual, counters, external, etc.) For each frame, the FPGA sends a programmable (1, 2 or 4) number of D/A codes from the generator's memory bank straight to the DAC. This transfer is done in latched mode, and the DAC is updated after all codes in a frame are sent. The generator continues this process, incrementing through the memory until it reaches the end of the buffer, or hits a programmable depth – at which point it will wrap back to the beginning of the buffer and continue operation. The generator can be paused, resumed or reset to the beginning of the memory bank at any time.

With the use of the memory block the D/A waveform generator can output consistent waveforms at a maximum frequency of 100KHz. There are four different input sources available for the D/A waveform generator: manual software trigger, counter 0 output, counters 1+2 output, and external trigger. The memory block also allows a programmable depth which when hit, will wrap and return to the beginning. The threshold ranges from 64 to 1024 and is programmable in multiples of 64.

14.2 Programming the D/A wave form generator

This section details how to program the D/A waveform generator through direct I/O without using the driver software. Please note that the D/A waveform generator is an enhanced feature and users must enabled enhanced features to access the feature. Details of the D/A waveform register map can be found at page 31.

There are 6 steps to programming the D/A waveform generator

1. **Enable enhanced features**
2. **Reset D/A waveform pointer**
3. **Latch D/A value**
4. **Store D/A values into buffer**
5. **Setup D/A wave form settings**
6. **Start D/A waveform generator**

14.3 Enable enhanced features

To enable enhanced features please consult page 32.

14.4 Reset D/A wave form pointer

Reset the D/A waveform pointer by accessing Page 5, Base + 15, bit 2. Writing a 1 to this bit and cause the pointer to start at the beginning, address 0.

14.5 Latch D/A value

Procedure for latching a D/A value is nearly identical to the formula in chapter 13 "Generating An Analog Output", page 44. The D/A value code must be computed for the desired voltage, from that obtain the LSB and MSB, add the channel number and set DAGEN bit. Write final LSB and MSB to registers 4 and 5. The only difference from the formula in the previous chapter is instead of setting the DASIM bit to 1, set DAGEN bit to 1. By setting the DAGEN bit to 1 the D/A value written will be latched to internal memory instead of the DAC chip.

14.6 Store D/A values into buffer

Once the D/A code is latched, it must be stored in the waveform buffer. Set Page to 5 and write the buffer address (0 to 1023) for the latched D/A value into Base + 12 and 13. When Base + 13 is written to, the latched D/A value at Base + 4 and 5 will be loaded and stored into the waveform memory. Both the D/A output code and D/A output channel would be stored.

14.7 Setup D/A wave form settings

D/A waveform settings include input source, number of code per frame, and threshold. Each can be set individually and in any combination.

There are four different input sources to choose from: manual/software trigger, counter 0 output, counters 1+2 output, and external trigger. Manual trigger should be used when the rate is slow or inconsistent and needs be controlled in software. Counter 0 output should be used when a consistent rate is desired and counter 1/2 is used for A/D interrupts. Counter 1+2 should be used when a consistent rate is desired and counter 0 is used for other interrupt functions, or if you want to synchronize the waveform generator to A/D interrupt functionality. External trigger should be used when an external signal is desired to generate D/A waveform. Input source is set by bits 0 and 1 on Page 5, Base + 14.

Number of code per frame determines the number of buffer values that will be output per frame. Each code is determined by the value set at its address.

For example, if the codes per frame option is set at 2, the first frame will output the codes at address 0 and 1, then 2 and 3, then 4 and 5 and so on. Number of code per frame is set by bits 2 and 3 on Page 5, Base + 14.

Threshold determines the number of code to output before the pointer starts over. The threshold must be set in multiples of 64 up to 1024. When the threshold is hit the pointer wraps around and start at the beginning. Threshold is set by bits 4, 5, 6, and 7 at Page 5, Base + 14.

14.8 Start D/A waveform generator

Initialize D/A waveform output by writing 1 to bit 0 at Page 5, Base + 15. The generator will continue to output the periodic waveform until you disable it.

15. AUTOCALIBRATION

Diamond-MM-32DX-AT features automatic calibration of both analog inputs and outputs. The potentiometers, which are subject to tampering, vibration, and maladjustment, have been completely eliminated. Instead, all calibration adjustments are performed using an octal 8-bit TrimDAC and precision, low-drift reference voltages on the board. The optimum TrimDAC values for each input range are stored in an EEPROM and recalled automatically on power up. As an upgrade from the DMM-32-AT, the DMM-32DX-AT now has the A/D autocalibration algorithm programmed into the dsPIC, featuring a faster, autonomous autocalibration.

To calibrate the board through software a calibration utility program and software driver function enables you to calibrate the analog inputs and outputs at any time for any range and store the settings in the EEPROM. This feature dramatically improves the accuracy and reliability of the board, since you can calibrate the board as often as desired without worrying about temperature or time drift.

On the analog outputs, the full-scale output range is programmable to any voltage up to 10V, and the board will calibrate to the programmed range. The analog outputs are fed back to the A/D converter so that they too can be calibrated without user intervention.

How it Works

The DMM-32DX-AT autocalibration circuit uses an octal 8-bit TrimDAC IC to provide small adjustments to the offset and gain at various points in the circuit. Four of the DACs are used for the A/D calibration, and the other four are used for the D/A. The 8-bit TrimDAC values are stored in an on-board EEPROM and are recalled automatically on power-up.

An on-board ultra-stable +5V reference chip with 5ppm offset drift is used as the voltage reference for all calibration operations. From this reference several intermediate values are derived that are used for the calibration. One is just under +5V, and one is just above 0V. These values are measured at the factory, and their values are stored in the on-board EEPROM for use by the calibration program. Note that the actual values of the reference signals does not matter, as long as they are stable, since the calibration routine knows the values and can adjust the calibration circuit to achieve them. An extra input multiplexor chip is used to feed the calibration voltages into the A/D circuit during the process.

For bipolar A/D calibration, first 0V is measured, and the TrimDAC is adjusted until the target A/D reading is achieved. For unipolar calibration, the voltage just above 0 is used as the first measurement value. Two TrimDAC channels are used for the offset. The first channel provides a coarse adjustment to bring the A/D readings into range, and then the second channel provides a fine adjustment for maximum accuracy. The use of both coarse and fine adjustments provides a wider range of total adjustment capability. The range of the fine adjustment exceeds the smallest change in the coarse adjustment, so there is no gap in the adjustment range.

After the offset is adjusted, the full-scale is adjusted in a similar manner. The reference value just under 5V is fed into the A/D, and two additional TrimDACs provide coarse and fine adjustments to achieve the target A/D near-full-scale reading.

Once the A/D is completely calibrated, the 16-bit (or 12-bit) D/A channels can be calibrated. Unlike the A/D circuit, which uses a single A/D for all input channels, the D/A circuit actually contains a single D/A converter for each of the 4 output channels. These channels are fed into the calibration multiplexor and the remaining 4 TrimDAC channels are used to calibrate them in a similar manner to the A/D. A single adjustment is used for the high reference, and both coarse and fine adjustments are used for the low reference.

The entire process takes about one second for each input range. Once it is complete, the board is ready to run. All 8 TrimDAC values are stored in the EEPROM so that the next time power is cycled to the board, the values will be loaded automatically.

How to perform autocalibration with software

The Universal Driver™ software provides two functions, dscADAutocal() and dscDAAutocal(), that can be called from within a user program to calibrate the board at any time. In addition, a standalone DOS program, DMM32CAL.EXE, is provided to enable calibration without requiring any programming.

How to perform autocalibration with dsPIC

The Diamond-MM-32DX-AT has an onboard microprocessor that can perform autocalibration for you, automatically. The microprocessor can be configured to trigger calibration because of temperature changes, or it can be manually triggered by software.

Temperature-triggered auto autocalibration is enabled by writing a 1 to Page 4, Base + 14, bit 3, ACREL. When this bit is set, the Autocal holdoff line is released. The dsPIC will monitor temperature changes and autocalibrate the board every 5'C.

If users want more control over auto autocalibration, they can set the ACHOLD bit to stop the temperature trigger (this is the default state), and use the ACTRIG bit to engage autocalibration when desired.

While the dsPIC is running auto autocalibration, Base + 4, bit 5, ACACT will be high. Users should monitor this bit whenever they have enabled auto autocalibration. While the dsPIC is autocalibrating, user software may not interact with the A/D circuit. When performing regular A/D functions users should set ACHOLD bit high to turn off auto autocalibration.

16. DIGITAL I/O OPERATION

Diamond-MM-32DX-AT contains two sets of digital I/O lines:

- An internal 82C55-type digital I/O circuit provides 24 digital I/O lines that emulate the function of Mode 0 of an 8255 chip. These lines are buffered to provide extra drive current in output mode and are available on digital I/O header J4 on the left side of the board.
- Analog I/O header J3 on the right side of the board contains 4 inputs and 3 outputs that can be used for general purpose DIO as long as they are not used for any special functions.

16.1 Main Digital I/O on J4: Internal 82C55 Circuit

The 82C55-type digital I/O circuit is accessed through page 1 at addresses Base + 12 through Base + 15. Address 0 on the chip is equivalent to address 12 in the register map, etc. Before performing any access to the digital I/O circuit, you must set the current page to page 1 with the miscellaneous control register at Base + 8 to ensure that the proper page is enabled. See page 21 for the format of this register. Note that writing page bits to the miscellaneous control register will not implement a board reset or interrupt reset operation as long as the two reset bits are left at 0. Also, writing a 1 to either reset bit in this register will not change the contents of the page bits.

The current page may be determined by reading the page bits at Base + 7 (see page 20).

This digital I/O circuit functions like an 82C55 in Mode 0, direct I/O, or Mode 1, latched I/O. In Mode 1, latch and acknowledge signals are provided. Each port A, B, and C can be programmed for input or output. Port C additionally can be split into two halves, with each half programmed for a different direction.

All 24 lines have 10KΩ resistors connected to them that can be configured for either pull-up or pull-down operation with jumper block J8. In addition, all lines are buffered by 74FCT245 line drivers between the controller chip and the I/O header. These line drivers change direction automatically in response to the control word written.

On power up, all ports are set to input mode and can be used as inputs immediately. Before using any port as an output, the port direction register must be programmed appropriately.

82C55 Circuit Register Map

Page 1, Base + 12 through Base + 15

Base + n	D7	D6	D5	D4	D3	D2	D1	D0
12, Read / Write	A7	A6	A5	A4	A3	A2	A1	A0
13, Read / Write	B7	B6	B5	B4	B3	B2	B1	B0
14, Read / Write	C7	C6	C5	C4	C3	C2	C1	C0
15, Write only	1	ModeC	ModeA	DirA	DirCH	ModeB	DirB	DirCL

A7 – A0 Digital I/O port A

B7 – B0 Digital I/O port B

C7 – C0 Digital I/O port C

Base + 15 Configuration register; see next page

16.2 Digital I/O Configuration Register

The direction control register is programmed by writing to Base + 15 using the format below. Once you have set the port directions with this register, you can read and write to the ports as desired.

Digital I/O Configuration Register: Page 1, Base + 15

Bit No.	7	6	5	4	3	2	1	0
Name	1	ModeC	ModeA	DirA	DirCH	ModeB	DirB	DirCL

Definitions:

1	Bit 7 must be set to 1 to indicate port mode set operation.
DirA	Direction control for bits A7 – A0: 0 = output, 1 = input
DirB	Direction control for bits B7 – B0: 0 = output, 1 = input
DirCL	Direction control for bits C3 – C0: 0 = output, 1 = input
DirCH	Direction control for bits C7 – C4: 0 = output, 1 = input

ModeA, ModeB, ModeC I/O Mode for each port, 0 or 1

Here is a list of common configuration register values:

Configuration Byte				
Hex	**Decimal**	**Port A**	**Port B**	**Port C (both halves)**
9B	155	Input	Input	Input
92	146	Input	Input	Output
99	153	Input	Output	Input
90	144	Input	Output	Output
8B	139	Output	Input	Input
82	130	Output	Input	Output
89	137	Output	Output	Input
80	128	Output	Output	Output

16.3 Mode 0 Digital I/O

This is the simpler of the two I/O modes and works well for most uses. In mode 0, the handshaking signals Latch and Ack are not used. When reading any port in input mode, the data at the I/O pins at the time of the read command will be returned.

16.4 Mode 1 Digital I/O With Handshaking

In Mode 1, a Latch input and an Acknowledge output signal are provided for handshaking operation. This allows the external circuit to tell the board when new input data is ready or when it has accepted the current output data, and it allows the board to tell the external circuit when it has read the current input data and when new output data is ready. Only Port A may be operated in Mode 1.

In all cases, the starting / resting conditions are Latch input = low and Acknowledge output = low.

Mode 1, Input, Non-Interrupt Operation

When the Latch input is brought high, Port A will latch the data on the I/O header pins, and the Acknowledge output will go high. The latches are D type flip-flops with positive-edge-triggered clocks, so just the rising edge of the Latch signal is used.

If a second rising edge occurs on the Latch input before the board reads the current data, the current data will be overwritten by whatever is then appearing on the input pins. The register's input latches are reset upon reading the register.

After the register has been read, the Acknowledge signal goes low to indicate that data has been accepted and new data may be latched.

Mode 1, Output, Non-Interrupt Operation

When data is written to Port A in mode 1 output, the Acknowledge output will go high, indicating that new data is available. When the Latch input is driven high by the external circuit, the Acknowledge output will go low. The external circuit may drive the Latch input low anytime after this happens.

Mode 1, Input, Interrupt Operation

When the Latch input is brought high, Port A will latch the data on the I/O header pins, the Acknowledge output will go high, and an interrupt request will be generated. When the data has been read from within the interrupt routine, the Acknowledge signal goes low to indicate that new data may be latched.

The interrupt routine is responsible for clearing the interrupt request signal from the board by writing a 1 to bit 3 of Base + 8.

Mode 1, Output, Interrupt Operation

When data is written to Port A in mode 1 output, the Acknowledge output will go high, indicating that new data is available. After the external devices latches the data, it drives the Latch input high, causing the Acknowledge output to go low and a new interrupt request to be generated. The interrupt routine then writes new data to Port A to restart the cycle.

Note that in this mode the program should write the first output value to Port A prior to the first interrupt being generated, so that the data is available to the external circuit before the first low-to-high Acknowledge transition. At the last interrupt, the program has no more data and simply terminates the operation.

16.5 Auxiliary Digital I/O on J3

J3 contains 3 digital outputs and 4 digital inputs that can be used for general purpose digital I/O or for A/D and counter/timer functions. The operation of these bits is controlled with various bits in two control registers.

Outputs

Ctr 2 Out / Dout 2 J3 pin 42

The function of this pin is determined by OUT2EN, Base + 10 bit 5:

1 Counter 2 output is routed to this pin

0 This pin is controlled by bit DOUT2 at Base + 1 bit 2

Dout 1 J3 pin 43

This pin is always the value written to DOUT1 at Base + 1 bit 1.

Ctr 0 Out / Dout 0 J3 pin 44

The function of this pin is determined by OUT0EN, Base + 10 bit 4:

1 Counter 0 output is routed to this pin

0 This pin is controlled by bit DOUT0 at Base + 1 bit 0

Inputs

Extclk / Din3 J3 pin 45

This signal may always be read at Base + 4 bit 3. It may function as an external clock to control A/D conversion timing when CLKEN = 1 and CLKSEL = 0 in Base + 9.

Extgate / Din2 J3 pin 46

This signal may always be read at Base + 4 bit 2. It may function as an external gate to enable and disable A/D conversions when GT12EN = 1 in Base + 10 bit 0.

Gate 0 / Din1 J3 pin 47

This signal may always be read at Base + 4 bit 1. It may function as an external gate for Counter 0 when GT0EN = 1 in Base + 10 bit 2. When used as a gate it is active high, meaning that Counter 0 will count as long as it is high and will not count when it is low.

Clk 0 / Din0 J3 pin 48

This signal may always be read at Base + 4 bit 0. It may function as an external clock for counter 0 when SRC0 = 0 in Base + 10 bit 1. When used as a clock for Counter 0, the rising edge is active.

17. COUNTER/TIMER OPERATION

17.1 Counter/Timer Features and Configuration Options

Diamond-MM-32DX-AT emulates an 82C54 counter/timer chip, providing 3 16-bit counter/timers.

Counters 1 and 2 are cascaded together to form a 32-bit counter/timer for use as a programmable A/D sampling clock. The output of counter 1 provides the input for counter 2, and the output of counter 2 is fed to the A/D triggering circuit as well as the I/O header J3. If not being used for A/D sampling, these counter/timers may be used for other functions. Counter/timer 0 is always available for user applications.

The inputs of the counter/timers are programmable, and the outputs may be routed to the I/O header under software control. The table below lists the key features of each counter/timer:

Counter/Timer Configuration Options

Counter	Input	Gate	Output
0	• 10MHz on-board • 10KHz on-board • Clk 0 / Din 0 (J3 pin 48)	• Gate 0 / Din 1 (J3 pin 47)	• Ctr 0 Out / Dout 0 (J3 pin 44)
1	• 10MHz • 100KHz	• Extgate / Din 2 (J3 pin 46)	• Not available to user
2	• Counter 1 out	• Extgate / Din 2 (J3 pin 46)	• Ctr 2 Out / Dout 2 (J3 pin 44) • Used internally for A/D sampling control

17.2 Counter/Timer Configuration

The counter/timer configuration is determined by the control register at Base + 10 described on page 23. Note that the outputs of counters 0 and 2 are routed to pins on I/O header J3 under software control rather than being hardwired.

Configuring the A/D sampling clock is done with the control register at Base + 9 described on page 22. Bit CLKEN selects whether the A/D hardware clocking is enabled, and if so, bit CLKSEL selects whether it is the output of counter/timer 2 or the external clock input at Extclk / Din3 on J3.

17.3 Counter/Timer Access and Programming

The emulated 8254 counter/timer chip is accessed through page 0 at addresses Base + 12 through Base + 15. Address 0 on the chip is equivalent to address 12 in the register map, etc. Before performing any access to the chip, you must set the current page to page 0 with the miscellaneous control register at Base + 8 to ensure that the proper page is enabled. See page 21 for the format of this register. Note that writing page bits to the miscellaneous control register will not implement a board reset or interrupt reset operation as long as the two reset bits are left at 0. Also, writing a 1 to either reset bit in this register will not change the contents of the page bits.

The current page may be determined by reading the page bits at Base + 7 (see page 20).

Once you write the proper page value, you can read and write to the 82C54 registers.

18. SPECIFICATIONS

Analog Inputs

Number of inputs	32 single-ended, 16 differential, or 16 SE and 8 DI	
A/D resolution	16 bits (1/65536 of full scale)	
Input ranges	Bipolar:	±10V, ±5V, ±2.5V, ±1.25V, ±0.625V
	Unipolar:	0 - 10V, 0 - 5V, 0 - 2.5V, 0 - 1.25V
Input bias current	100pA max	
Maximum input voltage	±10V for linear operation	
Overvoltage protection	±35V on any analog input without damage	
Input Impedance	10^13 ohms (10,000 Gohms) within normal input ranges	
Nonlinearity	±3LSB, no missing codes	
Conversion rate	250,000 samples per second max, single channel	
Conversion trigger	Software command, internal pacer clock, or external TTL signal	

Analog Outputs

Number of outputs	4	
D/A resolution	16 bits (1/4096 of full scale)	
Full-scale output ranges	Fixed Unipolar:	0 - 5V or 0 - 10V
	Fixed Bipolar:	±2.5V, ±5V or ±10V
	Programmable:	0 - 10V or ±10V in .01V steps
Output current	±5mA max per channel	
Settling time	6uS max to ±1/2 LSB	
Relative accuracy	±1 LSB	
No linearity	±1 LSB, monotonic	
Output reference	+5V ±.005V	

Autocalibration

Circuits calibrated	A/D (all 9 input ranges) and D/A
A/D error after calibration	±2LSB
D/A error after calibration	±1LSB

Digital I/O

Number of lines	24 using 8255-type circuit
Handshaking	Latch input, acknowledge output available in 8255 mode 1 configuration
Input voltage	Logic 0: 0.0V min, 0.8V max; Logic 1: 2.0V min, 5.0V max
Input current	±1μA max
Output voltage	Logic 0: 0.0V min, 0.33V max; Logic 1: 2.4V min (at 15mA load), 5.0V max
Output current	+15/-64mA max per line
Auxiliary DIO	4 inputs, 3 outputs, TTL compatible

Specifications continued on next page

Counter/Timers and Interrupts

A/D Pacer clock	32-bit down counter (2 82C54 counters cascaded)
Clock sources	10MHz on-board clock oscillator 100KHz derived frequency External signal
General purpose	16-bit down counter (1 82C54 counter)
Clock sources	10MHz on-board clock oscillator 10KHz derived frequency External signal
Interrupt triggers	End of A/D conversion Latch input on digital I/O header Timer 0 output

Connectors

J1	⇒ PC/104 8-bit bus
J2	⇒ PC/104 16-bit bus
J3	⇒ Analog I/O, Digital I/O
J4	⇒ Digital I/O, Serial Port

Bus Interface

Compliance	⇒ PC/104 Spec. Rev. 2.5 ⇒ Selectable 8/16-bit data bus ⇒ I/O-mapped bus slave device ⇒ DMA supported ⇒ PCB Dmensions in(mm) 3.550 (90.17) x 3.775 (95.89) x 0.062 (1.57)

General

Power supply	+5VDC ±10%
Current consumption	410mA typical
Operating temperature	-40°C to +85°C
Operating humidity	5% to 95% noncondensing
Weight	3.4oz / 96g
MTBF	58,574 hours
RoHS	Compliant